geoip2 Documentation

Release 3.0.0

Gregory Oschwald

Mar 28, 2020

Contents

Description

This package provides an API for the GeoIP2 web services and databases. The API also works with MaxMind's free GeoLite2 databases.

Installation

To install the `geoip2` module, type:

```
$ pip install geoip2
```

If you are not able to use pip, you may also use easy_install from the source directory:

```
$ easy_install .
```

2.1 Database Reader Extension

If you wish to use the C extension for the database reader, you must first install the libmaxminddb C API. Please see the instructions distributed with it.

IP Geolocation Usage

IP geolocation is inherently imprecise. Locations are often near the center of the population. Any location provided by a GeoIP2 database or web service should not be used to identify a particular address or household.

Usage

To use this API, you first create either a web service object with your MaxMind `account_id` and `license_key` or a database reader object with the path to your database file. After doing this, you may call the method corresponding to request type (e.g., `city` or `country`), passing it the IP address you want to look up.

If the request succeeds, the method call will return a model class for the end point you called. This model in turn contains multiple record classes, each of which represents part of the data returned by the web service.

If the request fails, the client class throws an exception.

Web Service Example

```
>>> import geoip2.webservice
>>>
>>> # This creates a Client object that can be reused across requests.
>>> # Replace "42" with your account ID and "license_key" with your license
>>> # key.
>>> client = geoip2.webservice.Client(42, 'license_key')
>>>
>>> # Replace "insights" with the method corresponding to the web service
>>> # that you are using, e.g., "country", "city".
>>> response = client.insights('128.101.101.101')
>>>
>>> response.country.iso_code
'US'
>>> response.country.name
'United States'
>>> response.country.names['zh-CN']
u''
>>>
>>> response.subdivisions.most_specific.name
'Minnesota'
>>> response.subdivisions.most_specific.iso_code
'MN'
>>>
>>> response.city.name
'Minneapolis'
>>>
>>> response.postal.code
'55455'
>>>
>>> response.location.latitude
44.9733
>>> response.location.longitude
-93.2323
>>>
```

```
>>> response.traits.network
IPv4Network('128.101.101.101/32')
```

Web Service Client Exceptions

For details on the possible errors returned by the web service itself, see http://dev.maxmind.com/geoip/geoip2/web-services for the GeoIP2 Precision web service docs.

If the web service returns an explicit error document, this is thrown as a `AddressNotFoundError`, `AuthenticationError`, `InvalidRequestError`, or `OutOfQueriesError` as appropriate. These all subclass `GeoIP2Error`.

If some other sort of error occurs, this is thrown as an `HTTPError`. This is thrown when some sort of unanticipated error occurs, such as the web service returning a 500 or an invalid error document. If the web service returns any status code besides 200, 4xx, or 5xx, this also becomes an `HTTPError`.

Finally, if the web service returns a 200 but the body is invalid, the client throws a `GeoIP2Error`.

Database Example

7.1 City Database

```
>>> import geoip2.database
>>>
>>> # This creates a Reader object. You should use the same object
>>> # across multiple requests as creation of it is expensive.
>>> reader = geoip2.database.Reader('/path/to/GeoLite2-City.mmdb')
>>>
>>> # Replace "city" with the method corresponding to the database
>>> # that you are using, e.g., "country".
>>> response = reader.city('128.101.101.101')
>>>
>>> response.country.iso_code
'US'
>>> response.country.name
'United States'
>>> response.country.names['zh-CN']
u''
>>>
>>> response.subdivisions.most_specific.name
'Minnesota'
>>> response.subdivisions.most_specific.iso_code
'MN'
>>>
>>> response.city.name
'Minneapolis'
>>>
>>> response.postal.code
'55455'
>>>
>>> response.location.latitude
44.9733
>>> response.location.longitude
```

```
-93.2323
>>>
>>> response.traits.network
IPv4Network('128.101.101.0/24')
>>>
>>> reader.close()
```

7.2 Anonymous IP Database

```
>>> import geoip2.database
>>>
>>> # This creates a Reader object. You should use the same object
>>> # across multiple requests as creation of it is expensive.
>>> reader = geoip2.database.Reader('/path/to/GeoIP2-Anonymous-IP.mmdb')
>>>
>>> response = reader.anonymous_ip('85.25.43.84')
>>>
>>> response.is_anonymous
True
>>> response.is_anonymous_vpn
False
>>> response.is_hosting_provider
False
>>> response.is_public_proxy
False
>>> response.is_tor_exit_node
True
>>> response.ip_address
'85.25.43.84'
>>> response.network
IPv4Network('85.25.43.0/24')
>>> reader.close()
```

7.3 ASN Database

```
>>> import geoip2.database
>>>
>>> # This creates a Reader object. You should use the same object
>>> # across multiple requests as creation of it is expensive.
>>> with geoip2.database.Reader('/path/to/GeoLite2-ASN.mmdb') as reader:
>>>     response = reader.asn('1.128.0.0')
>>>     response.autonomous_system_number
1221
>>>     response.autonomous_system_organization
'Telstra Pty Ltd'
```

7.4 Connection-Type Database

```
>>> import geoip2.database
>>>
>>> # This creates a Reader object. You should use the same object
>>> # across multiple requests as creation of it is expensive.
>>> reader = geoip2.database.Reader('/path/to/GeoIP2-Connection-Type.mmdb')
>>>
>>> response = reader.connection_type('128.101.101.101')
>>>
>>> response.connection_type
'Corporate'
>>> response.ip_address
'128.101.101.101'
>>> response.network
IPv4Network('128.101.101.101/24')
>>> reader.close()
```

7.5 Domain Database

```
>>> import geoip2.database
>>>
>>> # This creates a Reader object. You should use the same object
>>> # across multiple requests as creation of it is expensive.
>>> reader = geoip2.database.Reader('/path/to/GeoIP2-Domain.mmdb')
>>>
>>> response = reader.domain('128.101.101.101')
>>>
>>> response.domain
'umn.edu'
>>> response.ip_address
'128.101.101.101'
>>> reader.close()
```

7.6 Enterprise Database

```
>>> import geoip2.database
>>>
>>> # This creates a Reader object. You should use the same object
>>> # across multiple requests as creation of it is expensive.
>>> with geoip2.database.Reader('/path/to/GeoIP2-Enterprise.mmdb') as reader:
>>>
>>>     # Use the .enterprise method to do a lookup in the Enterprise database
>>>     response = reader.enterprise('128.101.101.101')
>>>
>>>     response.country.confidence
99
>>>     response.country.iso_code
'US'
>>>     response.country.name
'United States'
```

(continues on next page)

```
>>>     response.country.names['zh-CN']
u''
>>>
>>>     response.subdivisions.most_specific.name
'Minnesota'
>>>     response.subdivisions.most_specific.iso_code
'MN'
>>>     response.subdivisions.most_specific.confidence
77
>>>
>>>     response.city.name
'Minneapolis'
>>>     response.country.confidence
11
>>>
>>>     response.postal.code
'55455'
>>>
>>>     response.location.accuracy_radius
50
>>>     response.location.latitude
44.9733
>>>     response.location.longitude
-93.2323
>>>
>>>     response.traits.network
IPv4Network('128.101.101.0/24')
```

7.7 ISP Database

```
>>> import geoip2.database
>>>
>>> # This creates a Reader object. You should use the same object
>>> # across multiple requests as creation of it is expensive.
>>> reader = geoip2.database.Reader('/path/to/GeoIP2-ISP.mmdb')
>>>
>>> response = reader.isp('1.128.0.0')
>>>
>>> response.autonomous_system_number
1221
>>> response.autonomous_system_organization
'Telstra Pty Ltd'
>>> response.isp
'Telstra Internet'
>>> response.organization
'Telstra Internet'
>>> response.ip_address
'1.128.0.0'
>>> response.network
IPv4Network('1.128.0.0/16')
>>> reader.close()
```

Database Reader Exceptions

If the database file does not exist or is not readable, the constructor will raise a `FileNotFoundError` on Python 3 or an `IOError` on Python 2. If the IP address passed to a method is invalid, a `ValueError` will be raised. If the file is invalid or there is a bug in the reader, a `maxminddb.InvalidDatabaseError` will be raised with a description of the problem. If an IP address is not in the database, a `AddressNotFoundError` will be raised.

Values to use for Database or Dictionary Keys

We strongly discourage you from using a value from any ``names`` property as a key in a database or dictionaries.

These names may change between releases. Instead we recommend using one of the following:

- `geoip2.records.City` - `city.geoname_id`
- `geoip2.records.Continent` - `continent.code` or `continent.geoname_id`
- `geoip2.records.Country` and `geoip2.records.RepresentedCountry` - `country.iso_code` or `country.geoname_id`
- `geoip2.records.subdivision` - `subdivision.iso_code` or `subdivision.geoname_id`

CHAPTER 10

What data is returned?

While many of the models contain the same basic records, the attributes which can be populated vary between web service end points or databases. In addition, while a model may offer a particular piece of data, MaxMind does not always have every piece of data for any given IP address.

Because of these factors, it is possible for any request to return a record where some or all of the attributes are unpopulated.

The only piece of data which is always returned is the `ip_address` attribute in the `geoip2.records.Traits` record.

Integration with GeoNames

GeoNames offers web services and downloadable databases with data on geographical features around the world, including populated places. They offer both free and paid premium data. Each feature is uniquely identified by a `geoname_id`, which is an integer.

Many of the records returned by the GeoIP web services and databases include a `geoname_id` field. This is the ID of a geographical feature (city, region, country, etc.) in the GeoNames database.

Some of the data that MaxMind provides is also sourced from GeoNames. We source things like place names, ISO codes, and other similar data from the GeoNames premium data set.

Reporting Data Problems

If the problem you find is that an IP address is incorrectly mapped, please submit your correction to MaxMind.

If you find some other sort of mistake, like an incorrect spelling, please check the GeoNames site first. Once you've searched for a place and found it on the GeoNames map view, there are a number of links you can use to correct data ("move", "edit", "alternate names", etc.). Once the correction is part of the GeoNames data set, it will be automatically incorporated into future MaxMind releases.

If you are a paying MaxMind customer and you're not sure where to submit a correction, please contact MaxMind support for help.

Requirements

This code requires Python 2.7+ or 3.5+. Older versions are not supported. This library has been tested with CPython and PyPy.

The Requests HTTP library is also required. See <http://python-requests.org> for details.

CHAPTER 14

Versioning

The GeoIP2 Python API uses Semantic Versioning.

Support

Please report all issues with this code using the GitHub issue tracker

If you are having an issue with a MaxMind service that is not specific to the client API, please contact MaxMind support for assistance.

Modules

16.1 GeoIP2 Database Reader

class `geoip2.database.`**`Reader`**(*fileish*, *locales=None*, *mode=0*)
   GeoIP2 database Reader object.

   Instances of this class provide a reader for the GeoIP2 database format. IP addresses can be looked up using the `country` and `city` methods.

   The basic API for this class is the same for every database. First, you create a reader object, specifying a file name or file descriptor. You then call the method corresponding to the specific database, passing it the IP address you want to look up.

   If the request succeeds, the method call will return a model class for the method you called. This model in turn contains multiple record classes, each of which represents part of the data returned by the database. If the database does not contain the requested information, the attributes on the record class will have a `None` value.

   If the address is not in the database, an `geoip2.errors.AddressNotFoundError` exception will be thrown. If the database is corrupt or invalid, a `maxminddb.InvalidDatabaseError` will be thrown.

   `anonymous_ip`(*ip_address*)
      Get the AnonymousIP object for the IP address.

         Parameters **`ip_address`** – IPv4 or IPv6 address as a string.

         Returns *`geoip2.models.AnonymousIP`* object

   `asn`(*ip_address*)
      Get the ASN object for the IP address.

         Parameters **`ip_address`** – IPv4 or IPv6 address as a string.

         Returns *`geoip2.models.ASN`* object

   `city`(*ip_address*)
      Get the City object for the IP address.

         Parameters **`ip_address`** – IPv4 or IPv6 address as a string.

> > > **Returns** *geoip2.models.City* object

> **close**()
> > Closes the GeoIP2 database.

> **connection_type**(*ip_address*)
> > Get the ConnectionType object for the IP address.

> > > **Parameters** **ip_address** – IPv4 or IPv6 address as a string.

> > > **Returns** *geoip2.models.ConnectionType* object

> **country**(*ip_address*)
> > Get the Country object for the IP address.

> > > **Parameters** **ip_address** – IPv4 or IPv6 address as a string.

> > > **Returns** *geoip2.models.Country* object

> **domain**(*ip_address*)
> > Get the Domain object for the IP address.

> > > **Parameters** **ip_address** – IPv4 or IPv6 address as a string.

> > > **Returns** *geoip2.models.Domain* object

> **enterprise**(*ip_address*)
> > Get the Enterprise object for the IP address.

> > > **Parameters** **ip_address** – IPv4 or IPv6 address as a string.

> > > **Returns** *geoip2.models.Enterprise* object

> **isp**(*ip_address*)
> > Get the ISP object for the IP address.

> > > **Parameters** **ip_address** – IPv4 or IPv6 address as a string.

> > > **Returns** *geoip2.models.ISP* object

> **metadata**()
> > The metadata for the open database.

> > > **Returns** maxminddb.reader.Metadata object

16.2 WebServices Client API

This class provides a client API for all the GeoIP2 Precision web service end points. The end points are Country, City, and Insights. Each end point returns a different set of data about an IP address, with Country returning the least data and Insights the most.

Each web service end point is represented by a different model class, and these model classes in turn contain multiple record classes. The record classes have attributes which contain data about the IP address.

If the web service does not return a particular piece of data for an IP address, the associated attribute is not populated.

The web service may not return any information for an entire record, in which case all of the attributes for that record class will be empty.

16.2.1 SSL

Requests to the GeoIP2 Precision web service are always made with SSL.

class geoip2.webservice.**Client**(*account_id=None*, *license_key=None*, *host='geoip.maxmind.com'*, *locales=None*, *timeout=None*, *user_id=None*)

>Creates a new client object.

>It accepts the following required arguments:

>>**Parameters**

>>>- **account_id** – Your MaxMind account ID.

>>>- **license_key** – Your MaxMind license key.

>Go to https://www.maxmind.com/en/my_license_key to see your MaxMind account ID and license key.

>The following keyword arguments are also accepted:

>>**Parameters**

>>>- **host** – The hostname to make a request against. This defaults to "geoip.maxmind.com". In most cases, you should not need to set this explicitly.

>>>- **locales** – This is list of locale codes. This argument will be passed on to record classes to use when their name properties are called. The default value is ['en'].

>>> The order of the locales is significant. When a record class has multiple names (country, city, etc.), its name property will return the name in the first locale that has one.

>>> Note that the only locale which is always present in the GeoIP2 data is "en". If you do not include this locale, the name property may end up returning None even when the record has an English name.

>>> Currently, the valid locale codes are:

>>> – de – German

>>> – en – English names may still include accented characters if that is the accepted spelling in English. In other words, English does not mean ASCII.

>>> – es – Spanish

>>> – fr – French

>>> – ja – Japanese

>>> – pt-BR – Brazilian Portuguese

>>> – ru – Russian

>>> – zh-CN – Simplified Chinese.

city(*ip_address='me'*)

>Call GeoIP2 Precision City endpoint with the specified IP.

>>**Parameters ip_address** – IPv4 or IPv6 address as a string. If no address is provided, the address that the web service is called from will be used.

>>**Returns** *geoip2.models.City* object

country(*ip_address='me'*)

>Call the GeoIP2 Country endpoint with the specified IP.

> > **Parameters ip_address** – IPv4 or IPv6 address as a string. If no address is provided, the
> > address that the web service is called from will be used.
>
> > **Returns** *geoip2.models.Country* object

> **insights**(*ip_address='me'*)
> Call the GeoIP2 Precision: Insights endpoint with the specified IP.
>
> > **Parameters ip_address** – IPv4 or IPv6 address as a string. If no address is provided, the
> > address that the web service is called from will be used.
>
> > **Returns** *geoip2.models.Insights* object

16.3 Models

These classes provide models for the data returned by the GeoIP2 web service and databases.

The only difference between the City and Insights model classes is which fields in each record may be populated. See
http://dev.maxmind.com/geoip/geoip2/web-services for more details.

class geoip2.models.**ASN**(*raw*)
Model class for the GeoLite2 ASN.

This class provides the following attribute:

autonomous_system_number
The autonomous system number associated with the IP address.

> **Type** int

autonomous_system_organization
The organization associated with the registered autonomous system number for the IP address.

> **Type** unicode

ip_address
The IP address used in the lookup.

> **Type** unicode

network
The network associated with the record. In particular, this is the largest network where all of the fields
besides ip_address have the same value.

> **Type** ipaddress.IPv4Network or ipaddress.IPv6Network

class geoip2.models.**AnonymousIP**(*raw*)
Model class for the GeoIP2 Anonymous IP.

This class provides the following attribute:

is_anonymous
This is true if the IP address belongs to any sort of anonymous network.

> **Type** bool

is_anonymous_vpn
This is true if the IP address is registered to an anonymous VPN provider.

If a VPN provider does not register subnets under names associated with them, we will likely only flag
their IP ranges using the is_hosting_provider attribute.

> **Type** bool

is_hosting_provider
> This is true if the IP address belongs to a hosting or VPN provider (see description of is_anonymous_vpn attribute).
>
> > **Type** [bool](#)

is_public_proxy
> This is true if the IP address belongs to a public proxy.
>
> > **Type** [bool](#)

is_tor_exit_node
> This is true if the IP address is a Tor exit node.
>
> > **Type** [bool](#)

ip_address
> The IP address used in the lookup.
>
> > **Type** unicode

network
> The network associated with the record. In particular, this is the largest network where all of the fields besides ip_address have the same value.
>
> > **Type** [ipaddress.IPv4Network](#) or [ipaddress.IPv6Network](#)

class geoip2.models.**City**(*raw_response*, *locales=None*)
> Model for the GeoIP2 Precision: City and the GeoIP2 City database.

city
> City object for the requested IP address.
>
> > **Type** *[geoip2.records.City](#)*

continent
> Continent object for the requested IP address.
>
> > **Type** *[geoip2.records.Continent](#)*

country
> Country object for the requested IP address. This record represents the country where MaxMind believes the IP is located.
>
> > **Type** *[geoip2.records.Country](#)*

location
> Location object for the requested IP address.

maxmind
> Information related to your MaxMind account.
>
> > **Type** *[geoip2.records.MaxMind](#)*

registered_country
> The registered country object for the requested IP address. This record represents the country where the ISP has registered a given IP block in and may differ from the user's country.
>
> > **Type** *[geoip2.records.Country](#)*

represented_country
> Object for the country represented by the users of the IP address when that country is different than the country in country. For instance, the country represented by an overseas military base.
>
> > **Type** *[geoip2.records.RepresentedCountry](#)*

subdivisions
> Object (tuple) representing the subdivisions of the country to which the location of the requested IP address belongs.
>
> > **Type** *geoip2.records.Subdivisions*

traits
> Object with the traits of the requested IP address.
>
> > **Type** *geoip2.records.Traits*

class geoip2.models.**ConnectionType**(*raw*)
> Model class for the GeoIP2 Connection-Type.

> This class provides the following attribute:

connection_type
> The connection type may take the following values:
>
> > • Dialup
> >
> > • Cable/DSL
> >
> > • Corporate
> >
> > • Cellular
>
> Additional values may be added in the future.
>
> > **Type** unicode

ip_address
> The IP address used in the lookup.
>
> > **Type** unicode

network
> The network associated with the record. In particular, this is the largest network where all of the fields besides ip_address have the same value.
>
> > **Type** ipaddress.IPv4Network or ipaddress.IPv6Network

class geoip2.models.**Country**(*raw_response*, *locales=None*)
> Model for the GeoIP2 Precision: Country and the GeoIP2 Country database.

> This class provides the following attributes:

continent
> Continent object for the requested IP address.
>
> > **Type** *geoip2.records.Continent*

country
> Country object for the requested IP address. This record represents the country where MaxMind believes the IP is located.
>
> > **Type** *geoip2.records.Country*

maxmind
> Information related to your MaxMind account.
>
> > **Type** *geoip2.records.MaxMind*

registered_country
> The registered country object for the requested IP address. This record represents the country where the ISP has registered a given IP block in and may differ from the user's country.

>> **Type** *geoip2.records.Country*

represented_country
> Object for the country represented by the users of the IP address when that country is different than the country in `country`. For instance, the country represented by an overseas military base.

>> **Type** *geoip2.records.RepresentedCountry*

traits
> Object with the traits of the requested IP address.

>> **Type** *geoip2.records.Traits*

class geoip2.models.**Domain**(*raw*)
> Model class for the GeoIP2 Domain.

> This class provides the following attribute:

domain
> The domain associated with the IP address.

>> **Type** unicode

ip_address
> The IP address used in the lookup.

>> **Type** unicode

network
> The network associated with the record. In particular, this is the largest network where all of the fields besides ip_address have the same value.

>> **Type** ipaddress.IPv4Network or ipaddress.IPv6Network

class geoip2.models.**Enterprise**(*raw_response*, *locales=None*)
> Model for the GeoIP2 Enterprise database.

city
> City object for the requested IP address.

>> **Type** *geoip2.records.City*

continent
> Continent object for the requested IP address.

>> **Type** *geoip2.records.Continent*

country
> Country object for the requested IP address. This record represents the country where MaxMind believes the IP is located.

>> **Type** *geoip2.records.Country*

location
> Location object for the requested IP address.

maxmind
> Information related to your MaxMind account.

>> **Type** *geoip2.records.MaxMind*

registered_country
> The registered country object for the requested IP address. This record represents the country where the ISP has registered a given IP block in and may differ from the user's country.

>> **Type** *geoip2.records.Country*

represented_country
> Object for the country represented by the users of the IP address when that country is different than the country in country. For instance, the country represented by an overseas military base.
>
> > **Type** *geoip2.records.RepresentedCountry*

subdivisions
> Object (tuple) representing the subdivisions of the country to which the location of the requested IP address belongs.
>
> > **Type** *geoip2.records.Subdivisions*

traits
> Object with the traits of the requested IP address.
>
> > **Type** *geoip2.records.Traits*

class geoip2.models.**ISP**(*raw*)
> Model class for the GeoIP2 ISP.

This class provides the following attribute:

autonomous_system_number
> The autonomous system number associated with the IP address.
>
> > **Type** int

autonomous_system_organization
> The organization associated with the registered autonomous system number for the IP address.
>
> > **Type** unicode

isp
> The name of the ISP associated with the IP address.
>
> > **Type** unicode

organization
> The name of the organization associated with the IP address.
>
> > **Type** unicode

ip_address
> The IP address used in the lookup.
>
> > **Type** unicode

network
> The network associated with the record. In particular, this is the largest network where all of the fields besides ip_address have the same value.
>
> > **Type** ipaddress.IPv4Network or ipaddress.IPv6Network

class geoip2.models.**Insights**(*raw_response*, *locales=None*)
> Model for the GeoIP2 Precision: Insights web service endpoint.

city
> City object for the requested IP address.
>
> > **Type** *geoip2.records.City*

continent
> Continent object for the requested IP address.
>
> > **Type** *geoip2.records.Continent*

country
> Country object for the requested IP address. This record represents the country where MaxMind believes the IP is located.
>
> > **Type** *geoip2.records.Country*

location
> Location object for the requested IP address.

maxmind
> Information related to your MaxMind account.
>
> > **Type** *geoip2.records.MaxMind*

registered_country
> The registered country object for the requested IP address. This record represents the country where the ISP has registered a given IP block in and may differ from the user's country.
>
> > **Type** *geoip2.records.Country*

represented_country
> Object for the country represented by the users of the IP address when that country is different than the country in `country`. For instance, the country represented by an overseas military base.
>
> > **Type** *geoip2.records.RepresentedCountry*

subdivisions
> Object (tuple) representing the subdivisions of the country to which the location of the requested IP address belongs.
>
> > **Type** *geoip2.records.Subdivisions*

traits
> Object with the traits of the requested IP address.
>
> > **Type** *geoip2.records.Traits*

class geoip2.models.**SimpleModel**(*raw*)
> Provides basic methods for non-location models

> **network**
> > The network for the record

16.4 Records

class geoip2.records.**City**(*locales=None*, *confidence=None*, *geoname_id=None*, *names=None*, ***_*)
> Contains data for the city record associated with an IP address.
>
> This class contains the city-level data associated with an IP address.
>
> This record is returned by `city`, `enterprise`, and `insights`.
>
> Attributes:

> **confidence**
> > A value from 0-100 indicating MaxMind's confidence that the city is correct. This attribute is only available from the Insights end point and the GeoIP2 Enterprise database.
> >
> > > **Type** int

geoname_id
   The GeoName ID for the city.

   > **Type** [int](#)

name
   The name of the city based on the locales list passed to the constructor.

   > **Type** unicode

names
   A dictionary where the keys are locale codes and the values are names.

   > **Type** [dict](#)

class geoip2.records.**Continent**(*locales=None*, *code=None*, *geoname_id=None*, *names=None*, ***_)*
   Contains data for the continent record associated with an IP address.

   This class contains the continent-level data associated with an IP address.

   Attributes:

code
   A two character continent code like "NA" (North America) or "OC" (Oceania).

   > **Type** unicode

geoname_id
   The GeoName ID for the continent.

   > **Type** [int](#)

name
   Returns the name of the continent based on the locales list passed to the constructor.

   > **Type** unicode

names
   A dictionary where the keys are locale codes and the values are names.

   > **Type** [dict](#)

class geoip2.records.**Country**(*locales=None*, *confidence=None*, *geoname_id=None*, *is_in_european_union=False*, *iso_code=None*, *names=None*, ***_)*
   Contains data for the country record associated with an IP address.

   This class contains the country-level data associated with an IP address.

   Attributes:

confidence
   A value from 0-100 indicating MaxMind's confidence that the country is correct. This attribute is only available from the Insights end point and the GeoIP2 Enterprise database.

   > **Type** [int](#)

geoname_id
   The GeoName ID for the country.

   > **Type** [int](#)

is_in_european_union
   This is true if the country is a member state of the European Union.

   > **Type** [bool](#)

---

iso_code
> The two-character [ISO 3166-1](#) alpha code for the country.
>
> > **Type** unicode

name
> The name of the country based on the locales list passed to the constructor.
>
> > **Type** unicode

names
> A dictionary where the keys are locale codes and the values are names.
>
> > **Type** [dict](#)

class geoip2.records.**Location**(*average_income=None*, *accuracy_radius=None*, *latitude=None*, *longitude=None*, *metro_code=None*, *population_density=None*, *postal_code=None*, *postal_confidence=None*, *time_zone=None*, ***_*)

Contains data for the location record associated with an IP address.

This class contains the location data associated with an IP address.

This record is returned by city, enterprise, and insights.

Attributes:

average_income
> The average income in US dollars associated with the requested IP address. This attribute is only available from the Insights end point.
>
> > **Type** [int](#)

accuracy_radius
> The approximate accuracy radius in kilometers around the latitude and longitude for the IP address. This is the radius where we have a 67% confidence that the device using the IP address resides within the circle centered at the latitude and longitude with the provided radius.
>
> > **Type** [int](#)

latitude
> The approximate latitude of the location associated with the IP address. This value is not precise and should not be used to identify a particular address or household.
>
> > **Type** [float](#)

longitude
> The approximate longitude of the location associated with the IP address. This value is not precise and should not be used to identify a particular address or household.
>
> > **Type** [float](#)

metro_code
> The metro code of the location if the location is in the US. MaxMind returns the same metro codes as the [Google AdWords API](#).
>
> > **Type** [int](#)

population_density

> The estimated population per square kilometer associated with the IP address. This attribute is only available from the Insights end point.
>
> > **Type** [int](#)

> **time_zone**
> > The time zone associated with location, as specified by the IANA Time Zone Database, e.g., "America/New_York".
> >
> > > **Type** unicode

class geoip2.records.**MaxMind**(*queries_remaining=None*, ***_*)
> Contains data related to your MaxMind account.
>
> Attributes:
>
> **queries_remaining**
> > The number of remaining queries you have for the end point you are calling.
> >
> > > **Type** int

class geoip2.records.**PlaceRecord**(*locales=None*, *names=None*)
> All records with names subclass *PlaceRecord*.
>
> **name**
> > Dict with locale codes as keys and localized name as value.

class geoip2.records.**Postal**(*code=None*, *confidence=None*, ***_*)
> Contains data for the postal record associated with an IP address.
>
> This class contains the postal data associated with an IP address.
>
> This attribute is returned by city, enterprise, and insights.
>
> Attributes:
>
> **code**
> > The postal code of the location. Postal codes are not available for all countries. In some countries, this will only contain part of the postal code.
> >
> > > **Type** unicode
>
> **confidence**
> > A value from 0-100 indicating MaxMind's confidence that the postal code is correct. This attribute is only available from the Insights end point and the GeoIP2 Enterprise database.
> >
> > > **Type** int

class geoip2.records.**Record**
> All records are subclasses of the abstract class Record.

class geoip2.records.**RepresentedCountry**(*locales=None*, *confidence=None*, *geoname_id=None*, *is_in_european_union=False*, *iso_code=None*, *names=None*, *type=None*, ***_*)
> Contains data for the represented country associated with an IP address.
>
> This class contains the country-level data associated with an IP address for the IP's represented country. The represented country is the country represented by something like a military base.
>
> Attributes:
>
> **confidence**
> > A value from 0-100 indicating MaxMind's confidence that the country is correct. This attribute is only available from the Insights end point and the GeoIP2 Enterprise database.
> >
> > > **Type** int
>
> **geoname_id**
> > The GeoName ID for the country.
> >
> > > **Type** int

is_in_european_union
>   This is true if the country is a member state of the European Union.

>   > **Type** bool

iso_code
>   The two-character ISO 3166-1 alpha code for the country.

>   > **Type** unicode

name
>   The name of the country based on the locales list passed to the constructor.

>   > **Type** unicode

names
>   A dictionary where the keys are locale codes and the values are names.

>   > **Type** dict

type
>   A string indicating the type of entity that is representing the country. Currently we only return `military` but this could expand to include other types in the future.

>   > **Type** unicode

class `geoip2.records.`**Subdivision**(*locales=None*, *confidence=None*, *geoname_id=None*, *iso_code=None*, *names=None*, ***_*)
Contains data for the subdivisions associated with an IP address.

This class contains the subdivision data associated with an IP address.

This attribute is returned by `city`, `enterprise`, and `insights`.

Attributes:

confidence
>   This is a value from 0-100 indicating MaxMind's confidence that the subdivision is correct. This attribute is only available from the Insights end point and the GeoIP2 Enterprise database.

>   > **Type** int

geoname_id
>   This is a GeoName ID for the subdivision.

>   > **Type** int

iso_code
>   This is a string up to three characters long contain the subdivision portion of the ISO 3166-2 code.

>   > **Type** unicode

name
>   The name of the subdivision based on the locales list passed to the constructor.

>   > **Type** unicode

names
>   A dictionary where the keys are locale codes and the values are names

>   > **Type** dict

class `geoip2.records.`**Subdivisions**(*locales*, **subdivisions*)
A tuple-like collection of subdivisions associated with an IP address.

This class contains the subdivisions of the country associated with the IP address from largest to smallest.

For instance, the response for Oxford in the United Kingdom would have England as the first element and Oxfordshire as the second element.

This attribute is returned by `city`, `enterprise`, and `insights`.

most_specific
> The most specific (smallest) subdivision available.
>
> If there are no *Subdivision* objects for the response, this returns an empty *Subdivision*.
>
> > **Type** *Subdivision*

class `geoip2.records.`**`Traits`**(*autonomous_system_number=None, autonomous_system_organization=None, connection_type=None, domain=None, is_anonymous=False, is_anonymous_proxy=False, is_anonymous_vpn=False, is_hosting_provider=False, is_legitimate_proxy=False, is_public_proxy=False, is_satellite_provider=False, is_tor_exit_node=False, isp=None, ip_address=None, network=None, organization=None, prefix_len=None, static_ip_score=None, user_count=None, user_type=None, **_*)
Contains data for the traits record associated with an IP address.

This class contains the traits data associated with an IP address.

This class has the following attributes:

autonomous_system_number
> The autonomous system number associated with the IP address. This attribute is only available from the City and Insights web service end points and the GeoIP2 Enterprise database.
>
> > **Type** int

autonomous_system_organization
> The organization associated with the registered autonomous system number for the IP address. This attribute is only available from the City and Insights web service end points and the GeoIP2 Enterprise database.
>
> > **Type** unicode

connection_type
> The connection type may take the following values:
>
> * Dialup
>
> * Cable/DSL
>
> * Corporate
>
> * Cellular
>
> Additional values may be added in the future.
>
> This attribute is only available in the GeoIP2 Enterprise database.
>
> > **Type** unicode

domain
> The second level domain associated with the IP address. This will be something like "example.com" or "example.co.uk", not "foo.example.com". This attribute is only available from the City and Insights web service end points and the GeoIP2 Enterprise database.
>
> > **Type** unicode

ip_address
> The IP address that the data in the model is for. If you performed a "me" lookup against the web service, this will be the externally routable IP address for the system the code is running on. If the system is behind a NAT, this may differ from the IP address locally assigned to it.
>
> > **Type** unicode

is_anonymous
> This is true if the IP address belongs to any sort of anonymous network. This attribute is only available from GeoIP2 Precision Insights.
>
> > **Type** bool

is_anonymous_proxy
> This is true if the IP is an anonymous proxy. See http://dev.maxmind.com/faq/geoip#anonproxy for further details.
>
> > **Type** bool
>
> Deprecated since version 2.2.0: Use our our GeoIP2 Anonymous IP database instead.

is_anonymous_vpn
> This is true if the IP address is registered to an anonymous VPN provider.
>
> If a VPN provider does not register subnets under names associated with them, we will likely only flag their IP ranges using the is_hosting_provider attribute.
>
> This attribute is only available from GeoIP2 Precision Insights.
>
> > **Type** bool

is_hosting_provider
> This is true if the IP address belongs to a hosting or VPN provider (see description of is_anonymous_vpn attribute). This attribute is only available from GeoIP2 Precision Insights.
>
> > **Type** bool

is_legitimate_proxy
> This attribute is true if MaxMind believes this IP address to be a legitimate proxy, such as an internal VPN used by a corporation. This attribute is only available in the GeoIP2 Enterprise database.
>
> > **Type** bool

is_public_proxy
> This is true if the IP address belongs to a public proxy. This attribute is only available from GeoIP2 Precision Insights.
>
> > **Type** bool

is_satellite_provider
> This is true if the IP address is from a satellite provider that provides service to multiple countries.
>
> > **Type** bool
>
> Deprecated since version 2.2.0: Due to the increased coverage by mobile carriers, very few satellite providers now serve multiple countries. As a result, the output does not provide sufficiently relevant data for us to maintain it.

is_tor_exit_node
> This is true if the IP address is a Tor exit node. This attribute is only available from GeoIP2 Precision Insights.
>
> > **Type** bool

isp
> The name of the ISP associated with the IP address. This attribute is only available from the City and Insights web service end points and the GeoIP2 Enterprise database.
>
> > **Type** unicode

network
> The network associated with the record. In particular, this is the largest network where all of the fields besides ip_address have the same value.
>
> > **Type** ipaddress.IPv4Network or ipaddress.IPv6Network

organization
> The name of the organization associated with the IP address. This attribute is only available from the City and Insights web service end points and the GeoIP2 Enterprise database.
>
> > **Type** unicode

static_ip_score
> An indicator of how static or dynamic an IP address is. The value ranges from 0 to 99.99 with higher values meaning a greater static association. For example, many IP addresses with a user_type of cellular have a lifetime under one. Static Cable/DSL IPs typically have a lifetime above thirty.
>
> This indicator can be useful for deciding whether an IP address represents the same user over time. This attribute is only available from GeoIP2 Precision Insights.
>
> > **Type** float

user_count
> The estimated number of users sharing the IP/network during the past 24 hours. For IPv4, the count is for the individual IP. For IPv6, the count is for the /64 network. This attribute is only available from GeoIP2 Precision Insights.
>
> > **Type** int

user_type
> The user type associated with the IP address. This can be one of the following values:
>
> - business
> - cafe
> - cellular
> - college
> - content_delivery_network
> - dialup
> - government
> - hosting
> - library
> - military
> - residential
> - router
> - school
> - search_engine_spider
> - traveler

This attribute is only available from the Insights end point and the GeoIP2 Enterprise database.

> **Type** unicode

network
> The network for the record

16.5 Errors

exception `geoip2.errors.`**`AddressNotFoundError`**
> The address you were looking up was not found.

exception `geoip2.errors.`**`AuthenticationError`**
> There was a problem authenticating the request.

exception `geoip2.errors.`**`GeoIP2Error`**
> There was a generic error in GeoIP2.
>
> This class represents a generic error. It extends `RuntimeError` and does not add any additional attributes.

exception `geoip2.errors.`**`HTTPError`**(*message*, *http_status=None*, *uri=None*)
> There was an error when making your HTTP request.
>
> This class represents an HTTP transport error. It extends *GeoIP2Error* and adds attributes of its own.
>
> > **Variables**
> >
> > - **http_status** – The HTTP status code returned
> >
> > - **uri** – The URI queried

exception `geoip2.errors.`**`InvalidRequestError`**
> The request was invalid.

exception `geoip2.errors.`**`OutOfQueriesError`**
> Your account is out of funds for the service queried.

exception `geoip2.errors.`**`PermissionRequiredError`**
> Your account does not have permission to access this service.

Indices and tables

- genindex
- modindex
- search

copyright

(c) 2013-2019 by MaxMind, Inc.

license Apache License, Version 2.0

Python Module Index

g