
pygeons Documentation

Release 0.9.2

Michael Penkov

May 16, 2021

Contents

1	pygeons	3
1.1	Features	4
1.2	Credits	4
2	Installation	5
2.1	Stable release	5
2.2	From sources	5
2.3	Populating the Database	5
3	Usage	7
4	pygeons package	9
4.1	Module contents	9
5	Contributing	13
5.1	Types of Contributions	13
5.2	Get Started!	14
5.3	Pull Request Guidelines	15
5.4	Tips	15
6	Credits	17
6.1	Development Lead	17
6.2	Contributors	17
7	History	19
7.1	0.9.1 (2020-09-17)	19
7.2	0.9.0 (2020-09-11)	19
7.3	0.1.1 (2017-12-03)	19
7.4	0.1.0 (2017-11-26)	19
8	Indices and tables	21
	Python Module Index	23
	Index	25

Contents:

CHAPTER 1

pygeons

Geographical queries made simple.

- Free software: MIT license
- Documentation: <https://pygeons.readthedocs.io>.

Some examples:

```
>>> from pygeons.api import Country, find_cities
>>> Country('ivory coast')
Country('Ivory Coast')
>>> Country('côte d'ivoire')
Country('Ivory Coast')
>>> Country('civ')
Country('Ivory Coast')
>>> _.iso
'CI'
>>> Country('ivory coast').capital.name
'Yamoussoukro'
>>> Country('ivory coast').neighbors
[Country('Liberia'), Country('Ghana'), Country('Guinea'), Country('Burkina Faso'),
↪Country('Mali')]
>>>
>>> Country('us').cities['moscow']
City.gid(5601538, 'Moscow', 'US')
>>> Country('us').cities['moscow'].admin2
State.gid(5598264, 'ADM2', 'Latah County', 'US')
>>> Country('us').cities['moscow'].admin1
State.gid(5596512, 'ADM1', 'Idaho', 'US')
>>> Country('us').cities['moscow'].distance_to(Country('ru').cities['moscow'])
```

(continues on next page)

(continued from previous page)

```
8375.215117486288
>>>
>>> find_cities("oslo")[:2]
[City.gid(3143244, 'Oslo', 'NO'), City.gid(5040425, 'Oslo', 'US')]
```

1.1 Features

- Determine if a (city, state and country) combination corresponds to an existing place name
- Scrub (city, state, country) combinations
- Normalize city, state and country names to their canonical representations
- Frame queries in English as well as languages native to each particular country

1.2 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Stable release

To install pygeons, run this command in your terminal:

```
$ pip install pygeons
```

This is the preferred method to install pygeons, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for pygeons can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/mpenkov/pygeons
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/mpenkov/pygeons/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

2.3 Populating the Database

Pygeons requires data from [GeoNames.org](#). This data is free for sharing and adaptations as long as you abide by the [GeoNames license](#).

Pygeons includes scripts that download and import the data into a local sqlite3 DB. To download the data, run:

```
python -m pygeons.initialize
```

This will download approx. 500MB of data from geonames.org. Once the data is imported, the database will live under `.pygeons` in your home directory. Use the `PYGEONS_HOME` environment variable to modify this behavior. The data takes several GB, so make sure you have enough space.

CHAPTER 3

Usage

To use pygeons in a project:

```
import pygeons
```


4.1 Module contents

Top-level package for pygeons.

Implements low-level database structures and functions.

Expects you to call `connect()` before you do anything with the DB.

Expects the database to be initialized. If it is not, see `pygeons.initialize`.

By default, the database lives under `$HOME/.pygeons`. You can modify this behavior using the `PYGEONS_HOME` environment variable. You can also specify the subdirectory explicitly when you call `connect()`.

```
class pygeons.db.CountryInfo(iso, iso3, iso_numeric, fips, country, capital, area, popu-  
lation, continent, tld, currency_code, currency_name, phone,  
postal_code_format, postal_code_regex, languages, geonameid,  
neighbors, equivalent_fips_code)
```

Bases: tuple

area

Alias for field number 6

capital

Alias for field number 5

continent

Alias for field number 8

country

Alias for field number 4

currency_code

Alias for field number 10

currency_name

Alias for field number 11

equivalent_fips_code
Alias for field number 18

fips
Alias for field number 3

geonameid
Alias for field number 16

iso
Alias for field number 0

iso3
Alias for field number 1

iso_numeric
Alias for field number 2

languages
Alias for field number 15

neighbors
Alias for field number 17

phone
Alias for field number 12

population
Alias for field number 7

postal_code_format
Alias for field number 13

postal_code_regex
Alias for field number 14

tld
Alias for field number 9

class `pygeons.db.Geoname` (*geonameid, name, asciiname, alternatenames, latitude, longitude, feature_class, feature_code, country_code, cc2, admin1_code, admin2_code, admin3_code, admin4_code, population, elevation, dem, timezone, modification_date*)

Bases: `tuple`

admin1_code
Alias for field number 10

admin2_code
Alias for field number 11

admin3_code
Alias for field number 12

admin4_code
Alias for field number 13

alternatenames
Alias for field number 3

asciiname
Alias for field number 2

cc2
Alias for field number 9

country_code
Alias for field number 8

dem
Alias for field number 16

elevation
Alias for field number 15

feature_class
Alias for field number 6

feature_code
Alias for field number 7

geonameid
Alias for field number 0

latitude
Alias for field number 4

longitude
Alias for field number 5

modification_date
Alias for field number 18

name
Alias for field number 1

population
Alias for field number 14

timezone
Alias for field number 17

`pygeons.db.connect(subdir: str = '/home/docs/pygeons')` → None

`pygeons.db.country_info(name: str)` → `pygeons.db.CountryInfo`

```
>>> connect()
>>> i = country_info('ru')
>>> (i.country, i.population, i.currency_name)
('Russia', 144478050, 'Ruble')
```

`pygeons.db.get_alternatenames(geonameid: str)` → `List[Tuple[str, str]]`

`pygeons.db.select_geonames(subcommand: str, params: Iterable[Any])` → `List[pygeons.db.Geoname]`

`pygeons.db.select_geonames_ids(ids: Iterable[int], country_code: Optional[str] = None)` → `List[pygeons.db.Geoname]`

`pygeons.db.select_geonames_name(name: str)` → `List[pygeons.db.Geoname]`

Initialize data structures.

Downloads approx. 500MB of data from geonames.org.

`pygeons.initialize.build_trie(db_path: str, marisa_path: str)` → None

`pygeons.initialize.init_alternatename(db_path: str, fin: IO[str]) → None`

`pygeons.initialize.init_countryinfo(db_path: str) → None`

`pygeons.initialize.init_geoname(db_path: str, fin: IO[str]) → None`

`pygeons.initialize.init_postcode(db_path: str, fin: IO[str]) → None`

`pygeons.initialize.main()`

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/mpenkov/pygeons/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

5.1.4 Write Documentation

pygeons could always use more documentation, whether as part of the official pygeons docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/mpenkov/pygeons/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *pygeons* for local development.

1. Fork the *pygeons* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/pygeons.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv pygeons
$ cd pygeons/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 pygeons tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for the relevant versions of Python. Check https://travis-ci.org/mpenkov/pygeons/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_pygeons
```

To run documentation tests:

```
$ python -m doctest pygeons/*.py
```


CHAPTER 6

Credits

6.1 Development Lead

- Michael Penkov <misha.penkov@gmail.com>

6.2 Contributors

None yet. Why not be the first?

7.1 0.9.1 (2020-09-17)

- Added top-level *G'* collection to `pygeons.api`
- Implemented handling for alternative place names

7.2 0.9.0 (2020-09-11)

- Added new API
- Switched database backend from MongoDB to sqlite3
- Implemented import scripts in Python

7.3 0.1.1 (2017-12-03)

- First working release. Includes import scripts and source code.

7.4 0.1.0 (2017-11-26)

- First release on PyPI.

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`

p

`pygeons`, [9](#)

`pygeons.db`, [9](#)

`pygeons.initialize`, [11](#)

A

admin1_code (*pygeons.db.Geoname attribute*), 10
 admin2_code (*pygeons.db.Geoname attribute*), 10
 admin3_code (*pygeons.db.Geoname attribute*), 10
 admin4_code (*pygeons.db.Geoname attribute*), 10
 alternatenames (*pygeons.db.Geoname attribute*), 10
 area (*pygeons.db.CountryInfo attribute*), 9
 asciiname (*pygeons.db.Geoname attribute*), 10

B

build_trie() (*in module pygeons.initialize*), 11

C

capital (*pygeons.db.CountryInfo attribute*), 9
 cc2 (*pygeons.db.Geoname attribute*), 10
 connect() (*in module pygeons.db*), 11
 continent (*pygeons.db.CountryInfo attribute*), 9
 country (*pygeons.db.CountryInfo attribute*), 9
 country_code (*pygeons.db.Geoname attribute*), 11
 country_info() (*in module pygeons.db*), 11
 CountryInfo (*class in pygeons.db*), 9
 currency_code (*pygeons.db.CountryInfo attribute*), 9
 currency_name (*pygeons.db.CountryInfo attribute*), 9

D

dem (*pygeons.db.Geoname attribute*), 11

E

elevation (*pygeons.db.Geoname attribute*), 11
 equivalent_fips_code (*pygeons.db.CountryInfo attribute*), 9

F

feature_class (*pygeons.db.Geoname attribute*), 11
 feature_code (*pygeons.db.Geoname attribute*), 11
 fips (*pygeons.db.CountryInfo attribute*), 10

G

Geoname (*class in pygeons.db*), 10

geonameid (*pygeons.db.CountryInfo attribute*), 10
 geonameid (*pygeons.db.Geoname attribute*), 11
 get_alternatenames() (*in module pygeons.db*), 11

I

init_alternatename() (*in module pygeons.initialize*), 11
 init_countryinfo() (*in module pygeons.initialize*), 12
 init_geoname() (*in module pygeons.initialize*), 12
 init_postcode() (*in module pygeons.initialize*), 12
 iso (*pygeons.db.CountryInfo attribute*), 10
 iso3 (*pygeons.db.CountryInfo attribute*), 10
 iso_numeric (*pygeons.db.CountryInfo attribute*), 10

L

languages (*pygeons.db.CountryInfo attribute*), 10
 latitude (*pygeons.db.Geoname attribute*), 11
 longitude (*pygeons.db.Geoname attribute*), 11

M

main() (*in module pygeons.initialize*), 12
 modification_date (*pygeons.db.Geoname attribute*), 11

N

name (*pygeons.db.Geoname attribute*), 11
 neighbors (*pygeons.db.CountryInfo attribute*), 10

P

phone (*pygeons.db.CountryInfo attribute*), 10
 population (*pygeons.db.CountryInfo attribute*), 10
 population (*pygeons.db.Geoname attribute*), 11
 postal_code_format (*pygeons.db.CountryInfo attribute*), 10
 postal_code_regex (*pygeons.db.CountryInfo attribute*), 10
 pygeons (*module*), 9
 pygeons.db (*module*), 9

`pygeons.initialize(module)`, [11](#)

S

`select_geonames()` (*in module pygeons.db*), [11](#)

`select_geonames_ids()` (*in module pygeons.db*),
[11](#)

`select_geonames_name()` (*in module pygeons.db*),
[11](#)

T

`timezone` (*pygeons.db.Geoname attribute*), [11](#)

`tld` (*pygeons.db.CountryInfo attribute*), [10](#)