

pygeors

0.1

Generated by Doxygen 1.7.6.1

Sun Sep 1 2013 19:05:41

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Class Index	3
2.1	Class List	3
3	Namespace Documentation	5
3.1	geors Namespace Reference	5
3.1.1	Detailed Description	6
3.1.2	Function Documentation	6
3.1.2.1	area	6
3.1.2.2	distance	7
3.1.2.3	gcd	7
3.1.2.4	query	7
4	Class Documentation	9
4.1	geors.GeoLoc Class Reference	9
4.1.1	Detailed Description	10
4.1.2	Constructor & Destructor Documentation	10
4.1.2.1	__init__	10
4.1.3	Member Function Documentation	10
4.1.3.1	complete	10

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

[geors](#)

Name: geors.py Purpose: Geoinformation for DE based on opengeodb data and openstreetmap webservice Author: burger@burgerdev.de Created: 08.01.2013 Copyright: (c) Markus Döring 2013, plz.db is in the public domain (thanks to opengeodb.org) full text search results are covered by the - Database Contents License (DbCL) 1.0 Licence: GPL3, ODbL (see openstreetmap.org for details) 5

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

geors.GeoLoc	
Geographic location object	9

Chapter 3

Namespace Documentation

3.1 geors Namespace Reference

Name: geors.py Purpose: Geoinformation for DE based on opengeodb data and openstreetmap webservice Author: burger@burgerdev.de Created: 08.01.-2013 Copyright: (c) Markus Döring 2013, plz.db is in the public domain (thanks to opengeodb.org) full text search results are covered by the Database Contents License (DbCL) 1.0 Licence: GPL3, ODbL (see openstreetmap.org for details)

Classes

- class [GeoLoc](#)
geographic location object

Functions

- def [gcd](#)
great circle distance
- def [loc_factory](#)
get associative arrays aka dicts from sqlite3
- def [query](#)
query the geo database
- def [distance](#)
calculate distance between GeoLocs
- def [area](#)
get surrounding GeoLocs This function gets surrounding locations for a given location.

Variables

- tuple `_conn` = `sqlite3.connect(os.path.join(os.path.dirname(__file__), 'zipcode.-db'))`
- string `osm_useragent` = "pygeors - autocompletion"
openstreetmap user agent set this to your app's name
- `osm_email` = None
openstreetmap email address provide an email address for OSM to contact you

3.1.1 Detailed Description

Name: geors.py Purpose: Geoinformation for DE based on opengeodb data and openstreetmap webservice Author: burger@burgerdev.de Created: 08.01.-2013 Copyright: (c) Markus Döring 2013, plz.db is in the public domain (thanks to opengeodb.org) full text search results are covered by the Database Contents License (DbCL) 1.0 Licence: GPL3, ODbL (see openstreetmap.org for details)

3.1.2 Function Documentation

3.1.2.1 `def geors.area (loc, dist)`

get surrounding GeoLocs This function gets surrounding locations for a given location.

At the moment, there are a few restrictions to when this function will work as expected:
 * one of these is true * [GeoLoc.complete\(\)](#) finds a latitude/longitude pair (see there to check the requirements) * the input has its latitude/longitude pair [GeoLoc.latlon](#) set * the location is in Germany

Parameters

<i>loc</i>	a GeoLoc object to start with
<i>dist</i>	a distance in kilometres to search in

Returns

a list of GeoLocs (aka cities) within the specified radius of loc

usage (and doctests):

```
>>> area(None, 0)
>>>
>>> g = GeoLoc({'id': 1, 'zipcode': '87527', 'city': 'Sonthofen', 'latitude': 47.510178,
>>> L = area(g,0.1)
>>> len(L)
1
>>> L[0].zipcode
'87527'
```

3.1.2.2 def geors.distance (loc, locs)

calculate distance between GeoLocs

Parameters

<i>loc</i>	a GeoLoc object to start with
<i>locs</i>	either a GeoLoc object or a list of GeoLoc objects

Returns

a float or a list of floats (depending on the input) with the distance in kilometres

usage (and doctests):

```
>>> distance(None, None)
>>>
>>> distance(None, [None, None])
[None, None]
```

3.1.2.3 def geors.gcd (lat1, lon1, lat2, lon2, r = 6367.5)

great circle distance

compute the great circle distance between two latitude/longitude pairs

Parameters

<i>lat1</i>	first latitude
<i>lon1</i>	first longitude
<i>lat2</i>	second latitude
<i>lon2</i>	second longitude
<i>r</i>	radius (default: earth radius in km)

Returns

distance between the coordinates, units are the same as in the input

3.1.2.4 def geors.query (s)

query the geo database

Parameters

<i>s</i>	the query string (format: <city county>[,<state>] or <zipcode>)
----------	---

Returns

a [GeoLoc](#) object

Usage

```
>>> g = query('Hindelang')
>>> g.city
'Bad Hindelang'
>>> g.zipcode
'87541'
>>> g.state
'Bayern'
```

Chapter 4

Class Documentation

4.1 geors.GeoLoc Class Reference

geographic location object

Public Member Functions

- def `__init__`
the constructor
- def `__str__`
convert to string
- def `toString`
- def `complete`
look up [GeoLoc](#) information Complete the geographic information in this [GeoLoc](#) object.

Public Attributes

- `city`
the city (string, default: None)
- `zipcode`
the zip code (string, default: None)
- `county`
the county (string, default: None)
- `state`
the state (string, default: None)
- `country`
the country (string, default: "Deutschland")
- `countrycode`

the countrycode, in most cases TLD (string, default: "de")

- [latlon](#)

the city (tuple of two floats, default: None)

4.1.1 Detailed Description

geographic location object

This object represents a geographic location in terms of zip codes and city limits (as opposed to street addresses and the like). This object prefers German locations at the moment.

Basic Usage (and doctests):

```
>>> g = GeoLoc()
>>> g.zipcode = "87527"
>>> g.complete()
>>> g.city
'Sonthofen'

>>> g = GeoLoc()
>>> g.city = "Sonthofen"
>>> g.complete()
>>> g.zipcode
'87527'

>>> g = GeoLoc()
>>> g.latlon = (47.51, 10.29)
>>> g.complete()
>>> g.zipcode
'87527'
>>> g.city
'Sonthofen'
```

4.1.2 Constructor & Destructor Documentation

4.1.2.1 `def geors.GeoLoc.__init__(self, d=None)`

the constructor

Parameters

<i>d</i>	dict as returned by a query to geodb_zip (EXPERIMENTAL: dict as returned by a query to openstreetmap)
----------	---

4.1.3 Member Function Documentation

4.1.3.1 `def geors.GeoLoc.complete(self, useosm=False, deep=True)`

look up [GeoLoc](#) information Complete the geographic information in this [GeoLoc](#) object.

At the moment, this works if either * the zip code is set or * the latitude/longitude pair is set or * you are lucky with openstreetmap

Parameters

<i>useosm</i>	Specify if openstreetmap query should be sent (this feature is experimental!). *Don't forget to set geors.useragent and geors.email correctly!*
<i>deep</i>	Do you want precise results?

The documentation for this class was generated from the following file:

- geors.py