

---

# **Geogotchi Documentation**

***Release 0.2.2***

**Simon Pantzare**

October 08, 2012



# CONTENTS



Python library for working with [GeoNames](#) services.



# CONTENTS

## 1.1 Usage

```
>>> from geogotchi import Geogotchi
>>> lkpg = (58.411, 15.622)
>>> gg = Geogotchi(username="demo")
```

### 1.1.1 Find Nearby Toponyms

```
>>> gg.find_nearby_toponym(lkpg, radius=5, max_rows=3)
[{'countryCode': u'SE',
  'distance': 0.02928,
  'fcl': u'P',
  'fcode': u'PPLA',
  'geonameId': 2694762,
  'lat': 58.4108622585562,
  'lng': 15.62157154083252,
  'name': u'Link\xF6ping',
  'toponymName': u'Link\xF6ping'},
 ...]
```

### 1.1.2 Find Nearby Wikipedia Articles

Sorted by article rank and distance:

```
>>> gg.find_nearby_wikipedia(lkpg, radius=5, max_rows=3)
[{'countryCode': u'SE',
  'distance': 0.0401,
  'elevation': 58,
  'feature': u'city',
  'lang': u'en',
  'lat': 58.41083333333333,
  'lng': 15.62138888888889,
  'rank': 100,
  'summary': u'Link\xF6ping is a city in southern middle Sweden, with 104 232 inhabitants in 2010. ',
  'thumbnailImg': u'http://www.geonames.org/img/wikipedia/90000/thumb-89377-100.jpg',
  'title': u'Link\xF6ping',
  'wikipediaUrl': u'en.wikipedia.org/wiki/Link%C3%B6ping'},
 ...]
```

Sorted only by article rank (number of links pointing to it):

```
>>> gg.find_nearby_wikipedia(lkpg, radius=5, max_rows=3, rank_weight=1, distance_weight=0)
[{'countryCode': 'SE',
  'distance': 0.0401,
  'elevation': 58,
  'feature': 'city',
  'lang': 'en',
  'lat': 58.41083333333333,
  'lng': 15.62138888888889,
  'rank': 100,
  'summary': 'Linköping is a city in southern middle Sweden, with 104 232 inhabitants in 2010. ',
  'thumbnail': 'http://www.geonames.org/img/wikipedia/90000/thumb-89377-100.jpg',
  'title': 'Linköping',
  'wikipediaUrl': 'en.wikipedia.org/wiki/Linköping'},
 ...]
```

### 1.1.3 Hierarchies

Get all GeoNames higher up in the hierarchy of a place name:

```
>>> nearby = gg.find_nearby_place(lkpg)
>>> gg.get_hierarchy(nearby[0])
[{'adminName': '',
  'countryName': '',
  'fcl': 'L',
  'fclName': 'parks,area, ...',
  'fcode': 'AREA',
  'fcodeName': 'area',
  'geonameId': 6295630,
  'lat': 0,
  'lng': 0,
  'name': 'Earth',
  'population': 6814400000,
  'toponymName': 'Earth'},
 ...]
```

### 1.1.4 Text Search

Search for hotels in Sundsvall, Sweden:

```
>>> gg.search(q="sundsvall", feature_code="HTL")
[{'countryCode': 'SE',
  'fcl': 'S',
  'fcode': 'HTL',
  'geonameId': 6495350,
  'lat': 62.414,
  'lng': 17.349,
  'name': 'Scandic Sundsvall North',
  'toponymName': 'Scandic Sundsvall North'},
 ...]
```



## 1.2 API Reference

`class geogotchi.Geogotchi (username='demo')`

**find\_nearby\_place** (*latlon*, *\*\*kwargs*)

Find nearby populated place (reverse geocoding).

Does a “findNearbyPlaceNameJSON” API call behind the scenes.

### Parameters

- **latlon** – A latitude/longitude two-tuple.
- **radius** – Radius in km.
- **max\_rows** – Max number of rows.
- **style** – Verbosity (SHORT, MEDIUM, LONG or FULL).

**find\_nearby\_toponym** (*latlon*, *\*\*kwargs*)

Find nearby toponym (reverse geocoding).

Does a “findNearbyJSON” API call behind the scenes.

### Parameters

- **latlon** – A latitude/longitude two-tuple.
- **radius** – Radius in km.
- **max\_rows** – Max number of rows.
- **style** – Verbosity (SHORT, MEDIUM, LONG or FULL).

**find\_nearby\_wikipedia** (*latlon*, *rank\_weight=1.0*, *distance\_weight=1.0*, *\*\*kwargs*)

Find nearby Wikipedia entries (reverse geocoding).

Does a “findNearbyWikipediaJSON” API call behind the scenes. Results are sorted in descending order.

### Parameters

- **latlon** – A latitude/longitude two-tuple.
- **radius** – Radius in km.
- **max\_rows** – Max number of rows.
- **rank\_weight** – Weight of rank in sorting, between 0.0 and 1.0.
- **distance\_weight** – Weight of distance in sorting, between 0.0 and 1.0.
- **lang** – Language code.

**get\_hierarchy** (*geoname*)

Returns all GeoNames higher up in the hierarchy of a place name.

**Parameters** **geoname** – A dict with a “geonameId” key or an integer.

**search** (*\*\*kwargs*)

Perform a search.

One of *q*, *name* and *name\_equals* must be given. Non-unicode string arguments are assumed to be UTF-8 encoded.

### Parameters

- **q** – Query over all attributes of a place.

- **name** – Query place name only.
- **name\_equals** – Query exact place name.
- **max\_rows** – Max number of rows.
- **start\_row** – Used for paging results. If you want to get results 30 to 40, use `start_row=30` and `max_rows=10`.
- **country** – ISO-3166 country code.
- **country\_bias** – Records from the country bias are listed first.
- **continent\_code** – Restricts the search for a toponym of the given continent (AF, AS, EU, NA, OC, SA, AN).
- **feature\_class** – [GeoNames feature class](#)
- **feature\_code** – [GeoNames feature code](#)
- **lang** – ISO-636 2-letter language code (en, de, fr, se...)
- **style** – Verbosity (SHORT, MEDIUM, LONG, FULL)
- **operator** – AND (default) or OR.
- **fuzzy** – Defines the fuzzyness of the search terms. Float between 0.0 and 1.0 (default 1.0).

# EXTERNAL RESOURCES

- [GeoNames feature classes and codes \(statistics\)](#)



# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*



# PYTHON MODULE INDEX

## g

geogotchi, ??