

# Package ‘geonames’

February 19, 2019

**Encoding** UTF-8

**Type** Package

**Title** Interface to the ``Geonames'' Spatial Query Web Service

**Version** 0.999

**Date** 2019-02-17

**Author** Barry Rowlingson

**Maintainer** Barry Rowlingson <b.rowlingson@gmail.com>

**Depends** R (>= 2.2.0)

**Imports** rjson

**Description** The web service at <<https://www.geonames.org/>> provides a number of spatial data queries, including administrative area hierarchies, city locations and some country postal code queries. A (free) username is required and rate limits exist.

**License** GPL-3

**LazyLoad** yes

**BugReports** <https://github.com/ropensci/geonames/issues>

**RoxygenNote** 6.1.0

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-02-19 12:30:15 UTC

## R topics documented:

geonames . . . . .	2
GNcities . . . . .	3
GNcountryCode . . . . .	3
GNcountryInfo . . . . .	4
GNcountrySubdivision . . . . .	5
GNearthquakes . . . . .	5

GNfindNearby . . . . .	6
GNfindNearbyPlaceName . . . . .	7
GNfindNearbyPostalCodes . . . . .	8
GNfindNearbyStreets . . . . .	8
GNfindNearByWeather . . . . .	9
GNfindNearbyWikipedia . . . . .	10
GNfindNearestAddress . . . . .	11
GNfindNearestIntersection . . . . .	11
GNgtopo30 . . . . .	12
GNneighbourhood . . . . .	13
GNpostalCodeCountryInfo . . . . .	13
GNpostalCodeLookup . . . . .	14
GNpostalCodeSearch . . . . .	15
GNsearch . . . . .	15
GNsrtm3 . . . . .	16
GNtimezone . . . . .	17
GNweather . . . . .	18
GNweatherIcao . . . . .	18
GNwikipediaBoundingBox . . . . .	19
GNwikipediaSearch . . . . .	20
hierarchy . . . . .	21
<b>Index</b>	<b>22</b>

---

geonames

*Query the geonames web API for geographic data*

---

## Description

www.geonames.org is a service where you can query for global geographic data such as administrative areas, populated places, weather data etc.

## Details

The functions in this package are mostly thin wrappers to the API calls documented at the geonames web services overview <http://www.geonames.org/export/ws-overview.html>.

A set of example calls are supplied in a file with the package. Once you have set your geonames username with `options(geonamesUsername="myusernamehere")` you can run this with `source(system.file("tests","testing.R",package="geonames"),echo=TRUE)`

---

GNcities	<i>find cities</i>
----------	--------------------

---

**Description**

Search for cities

**Usage**

```
GNcities(north, east, south, west, lang = "en", maxRows = 10)
```

**Arguments**

north	north bound
east	east bound
south	south bound
west	west bount
lang	language code
maxRows	max number of records to return

**Details**

find cities

API doc for GNcities is at <http://www.geonames.org/export/JSON-webservices.html#citiesJSON>

**Value**

city records

**Author(s)**

Barry Rowlingson

---

GNcountryCode	<i>country code for location</i>
---------------	----------------------------------

---

**Description**

Get country code

**Usage**

```
GNcountryCode(lat, lng, lang = "", radius = "")
```

**Arguments**

lat	latitude
lng	longitude
lang	language code
radius	radius size

**Details**

country code for location

API doc for GNcountryCode is at <http://www.geonames.org/export/web-services.html#countrycode>

**Value**

country record

**Author(s)**

Barry Rowlingson

---

GNcountryInfo	<i>country info</i>
---------------	---------------------

---

**Description**

Get country info

**Usage**

```
GNcountryInfo(country = "", lang = "")
```

**Arguments**

country	country code
lang	language code

**Details**

country info

API doc for GNcountryInfo is at <http://www.geonames.org/export/web-services.html#countryInfo>

**Value**

country record info

**Author(s)**

Barry Rowlingson

---

GNcountrySubdivision    *country code and subdivision*

---

**Description**

country code and admin subdivision

**Usage**

```
GNcountrySubdivision(lat, lng, lang = "en", radius = "",  
maxRows = 10)
```

**Arguments**

lat	latitude
lng	longitude
lang	language code
radius	search radius
maxRows	max number of returned records

**Details**

looks up country and admin subdivisions

API doc for GNcountrySubdivision is at <http://www.geonames.org/export/web-services.html#countrysubdiv>

**Value**

iso country code

**Author(s)**

Barry Rowlingson

---

GNearthquakes    *recent earthquakes*

---

**Description**

recent earthquakes

**Usage**

```
GNearthquakes(north, east, south, west, date, minMagnitude, maxRows = 10)
```

**Arguments**

north	north bound
east	east bound
south	south bound
west	west bound
date	optional date
minMagnitude	optional minimal magnitude
maxRows	max records to return

**Details**

get recent earthquakes in a region

API doc for GNearthquakes is at <http://www.geonames.org/export/JSON-webservices.html#earthquakesJSON>

**Value**

earthquake records

**Author(s)**

Barry Rowlingson

**Examples**

```
## Not run:  
GNearthquakes(north=44.1, south=-9.9, east=-22.4, west=55.2)  
  
## End(Not run)
```

---

GNfindNearby	<i>nearby search</i>
--------------	----------------------

---

**Description**

find nearby entities

**Usage**

```
GNfindNearby(...)
```

**Arguments**

... search parameters

**Details**

nearby search

API doc for GNfindNearby is at <http://www.geonames.org/export/web-services.html#findNearby>

**Value**

matched records

**Author(s)**

Barry Rowlingson

---

GNfindNearbyPlaceName *populated place search*

---

**Description**

find nearby populated place

**Usage**

```
GNfindNearbyPlaceName(lat, lng, radius = "", maxRows = "10",  
style = "MEDIUM")
```

**Arguments**

lat	latitude
lng	longitude
radius	search radius
maxRows	max records returned
style	verbosity of record

**Details**

search for populated places

API doc for GNfindNearbyPlaceName is at <http://www.geonames.org/export/web-services.html#findNearbyPlaceName>

**Value**

nearby populated place records

**Author(s)**

Barry Rowlingson

GNfindNearbyPostalCodes

*find postal code*

---

### Description

find postal code by lat long or code

### Usage

```
GNfindNearbyPostalCodes(...)
```

### Arguments

... search parameters, see geonames web docs for details

### Details

find postal code

API doc for GNfindNearbyPostalCodes is at <http://www.geonames.org/export/web-services.html#findNearbyPostalCodes>

### Value

postal code records

### Author(s)

Barry Rowlingson

---

GNfindNearbyStreets *nearby street finding*

---

### Description

find nearby streets (US only)

### Usage

```
GNfindNearbyStreets(lat, lng)
```

### Arguments

lat	latitude
lng	longitude



**Details**

for a lat-long, find nearby US streets

API doc for GNfindNearbyStreets is at <http://www.geonames.org/maps/us-reverse-geocoder.html#findNearbyStreets>

**Value**

street records

**Author(s)**

Barry Rowlingson

---

GNfindNearByWeather    *weather at location*

---

**Description**

get weather at location

**Usage**

GNfindNearByWeather(lat, lng)

**Arguments**

lat	latitude
lng	longitude

**Details**

get weather

API doc for GNfindNearByWeather is at <http://www.geonames.org/export/JSON-webservices.html#findNearByWeatherJSON>

**Value**

weather record

**Note**

check capitalisation of 'NearBy'

**Author(s)**

Barry Rowlingson

**Examples**

```
## Not run:  
GNfindNearByWeather(57,-2)  
  
## End(Not run)
```

---

GNfindNearbyWikipedia *nearby wikipedia entries*

---

**Description**

find nearby wikipedia entries

**Usage**

```
GNfindNearbyWikipedia(...)
```

**Arguments**

... see [geonames.org](http://geonames.org) documentation

**Details**

search wikipedia entries by lat/lng or location name parameters

API doc for GNfindNearbyWikipedia is at <http://www.geonames.org/export/wikipedia-webservice.html#findNearbyWikipedia>

**Value**

wikipedia entries

**Author(s)**

Barry Rowlingson

**Examples**

```
## Not run:  
GNfindNearbyWikipedia(postalcode=8775, country="CH", radius=10)  
  
## End(Not run)
```

---

GNfindNearestAddress *nearest address*

---

**Description**

find nearest street and address

**Usage**

GNfindNearestAddress(lat, lng)

**Arguments**

lat	latitude
lng	longitude

**Details**

search US for nearest street and address

API doc for GNfindNearestAddress is at <http://www.geonames.org/maps/us-reverse-geocoder.html#findNearestAddress>

**Value**

address record

**Author(s)**

Barry Rowlingson

---

GNfindNearestIntersection  
*nearest intersection*

---

**Description**

search US for nearest intersection

**Usage**

GNfindNearestIntersection(lat, lng)

**Arguments**

lat	latitude
lng	longitude

**Details**

finds nearest intersection

API doc for GNfindNearestIntersection is at <http://www.geonames.org/maps/us-reverse-geocoder.html#findNearestIntersection>

**Value**

intersection record

**Author(s)**

Barry Rowlingson

---

GNgtopo30	<i>topo30 height</i>
-----------	----------------------

---

**Description**

height from topo30

**Usage**

```
GNgtopo30(lat, lng)
```

**Arguments**

lat	latitude
lng	longitude

**Details**

get height from topo30 data

API doc for GNgtopo30 is at <http://www.geonames.org/export/web-services.html#gtopo30>

**Value**

height record

**Author(s)**

Barry Rowlingson

**Examples**

```
## Not run:  
GNgtopo30(lat=54, lng=-1)  
  
## End(Not run)
```

---

GNneighbourhood      *neighbourhood*

---

**Description**

find neighbourhood

**Usage**

GNneighbourhood(lat, lng)

**Arguments**

lat	latitude
lng	longitude

**Details**

find neighbourhood

API doc for GNneighbourhood is at <http://www.geonames.org/export/web-services.html#neighbourhood>

**Value**

neighbourhood records

**Author(s)**

Barry Rowlingson

---

GNpostalCodeCountryInfo  
*postal code info*

---

**Description**

countries with postal code info

**Usage**

GNpostalCodeCountryInfo()

**Details**

list countries with postal code info

API doc for GNpostalCodeCountryInfo is at <http://www.geonames.org/export/web-services.html#postalCodeCountryInfo>

**Value**

list of countries with postal codes on geonames

**Author(s)**

Barry Rowlingson

---

GNpostalCodeLookup     *postal code lookup*

---

**Description**

postal code lookup

**Usage**

```
GNpostalCodeLookup(...)
```

**Arguments**

...            parameters

**Details**

postal code lookup

API doc for GNpostalCodeLookup is at <http://www.geonames.org/export/web-services.html#postalCodeLookupJSON>

**Value**

list of places for a given input postal code

**Author(s)**

Barry Rowlingson

---

GNpostalCodeSearch	<i>postal code search</i>
--------------------	---------------------------

---

**Description**

search for postal code

**Usage**

```
GNpostalCodeSearch(...)
```

**Arguments**

... search parameters

**Details**

full text search for postal codes

API doc for GNpostalCodeSearch is at <http://www.geonames.org/export/web-services.html#postalCodeSearch>

**Value**

postal code record

**Author(s)**

Barry Rowlingson

---

GNsearch	<i>search geonames</i>
----------	------------------------

---

**Description**

search geonames

**Usage**

```
GNsearch(...)
```

**Arguments**

... search parameters

**Details**

general search call

API doc for GNsearch is at <http://www.geonames.org/export/geonames-search.html>

**Value**

matched records

**Author(s)**

Barry Rowlingson

**Examples**

```
## Not run:  
# Find places called Lancaster and 'geocode' the result  
(lanc_df <- GNsearch(name = "Lancaster", country = "UK"))  
lanc_coords <- lanc_df[1, c("lng", "lat")]  
  
## End(Not run)
```

---

GNSrstm3

*srtm3 height*

---

**Description**

height from srtm3 data

**Usage**

```
GNSrstm3(lat, lng)
```

**Arguments**

lat	latitude
lng	longitude

**Details**

get srtm3 height

API doc for GNSrstm3 is at <http://www.geonames.org/export/web-services.html#srtm3>

**Value**

height record



**Author(s)**

Barry Rowlingson

**Examples**

```
## Not run:  
GNSrtm3(lat=54, lng=-1)  
  
## End(Not run)
```

---

GNtimezone	<i>timezone for location</i>
------------	------------------------------

---

**Description**

get timezone

**Usage**

```
GNtimezone(lat, lng, radius = 0)
```

**Arguments**

lat	latitude
lng	longitude
radius	search radius

**Details**

timezone for location

API doc for GNtimezone is at <http://www.geonames.org/export/web-services.html#timezone>

**Value**

time zone record

**Author(s)**

Barry Rowlingson

GNweather

*weather stations in box*

---

**Description**

weather stations in region

**Usage**

GNweather(north, east, south, west, maxRows = 10)

**Arguments**

north	north bound
east	east bound
south	south bound
west	west bound
maxRows	max records to return

**Details**

get weather stations in region with latest readings

**Value**

weather records

**Author(s)**Barry Rowlingson

---

GNweatherIcao

*ICAO weather station data*

---

**Description**

weather record from ICAO station

**Usage**

GNweatherIcao(ICA0)

**Arguments**

ICA0	ICAO code
------	-----------

**Details**

get most recent ICAO station data

API doc for GNweatherIcao is at <http://www.geonames.org/export/JSON-webservices.html#weatherIcaoJSON>

**Value**

weather record

**Author(s)**

Barry Rowlingson

---

GNwikipediaBoundingBox

*wikipedia articles in a box*

---

**Description**

wikipedia articles in bounding box

**Usage**

```
GNwikipediaBoundingBox(...)
```

**Arguments**

```
...           parameters (north, south, east, west etc.)
```

**Details**

find articles in a box

API doc for GNwikipediaBoundingBox is at <http://www.geonames.org/export/wikipedia-webservice.html#wikipediaBoundingBox>

**Value**

wikipedia records

**Author(s)**

Barry Rowlingson

**Examples**

```
## Not run:  
GNwikipediaBoundingBox(north=44.1, south=-9.9, east=-22.4, west=55.2)  
  
## End(Not run)
```

GNwikipediaSearch      *search wikipedia*

---

### **Description**

wikipedia fulltext search

### **Usage**

```
GNwikipediaSearch(q, maxRows = 10)
```

### **Arguments**

q	search string
maxRows	maximum returned records

### **Details**

find geolocated articles in wikipedia

API doc for GNwikipediaSearch is at <http://www.geonames.org/export/wikipedia-webservice.html#wikipediaSearch>

### **Value**

wikipedia entries

### **Author(s)**

Barry Rowlingson

### **Examples**

```
## Not run:  
GNwikipediaSearch("london")  
  
## End(Not run)
```

---

hierarchy	<i>Admin area hierarchy</i>
-----------	-----------------------------

---

**Description**

See <http://www.geonames.org/export/ws-overview.html> for a full description of valid arguments and return values

**Usage**

GNchildren(geonameId, ...)

GNhierarchy(geonameId, ...)

GNsiblings(geonameId, ...)

GNneighbours(geonameId, ...)

**Arguments**

geonameId	a geonames ID value
...	other parameters to pass to geonames

**Details**

API doc for GNchildren is at <http://www.geonames.org/export/place-hierarchy.html#children>

API doc for GNhierarchy is at <http://www.geonames.org/export/place-hierarchy.html#hierarchy>

API doc for GNsiblings is at <http://www.geonames.org/export/place-hierarchy.html#siblings>

API doc for GNneighbours is at <http://www.geonames.org/export/place-hierarchy.html#neighbours>

# Index

geonames, [2](#)  
geonames-package (geonames), [2](#)  
GNchildren (hierarchy), [21](#)  
GNcities, [3](#)  
GNcountryCode, [3](#)  
GNcountryInfo, [4](#)  
GNcountrySubdivision, [5](#)  
GNearthquakes, [5](#)  
GNfindNearby, [6](#)  
GNfindNearbyPlaceName, [7](#)  
GNfindNearbyPostalCodes, [8](#)  
GNfindNearbyStreets, [8](#)  
GNfindNearByWeather, [9](#)  
GNfindNearbyWikipedia, [10](#)  
GNfindNearestAddress, [11](#)  
GNfindNearestIntersection, [11](#)  
GNgtopo30, [12](#)  
GNhierarchy (hierarchy), [21](#)  
GNneighbourhood, [13](#)  
GNneighbours (hierarchy), [21](#)  
GNpostalCodeCountryInfo, [13](#)  
GNpostalCodeLookup, [14](#)  
GNpostalCodeSearch, [15](#)  
GNsearch, [15](#)  
GNsiblings (hierarchy), [21](#)  
GNsrtm3, [16](#)  
GNtimezone, [17](#)  
GNweather, [18](#)  
GNweatherIcao, [18](#)  
GNwikipediaBoundingBox, [19](#)  
GNwikipediaSearch, [20](#)

hierarchy, [21](#)