Package 'nominatimlite'

October 13, 2022

Type Package

Title Interface with 'Nominatim' API Service

Version 0.1.6

Description Lite interface for getting data from 'OSM' service 'Nominatim' <https://nominatim.org/release-docs/latest/>. Extract coordinates from addresses, find places near a set of coordinates, search for amenities and return spatial objects on 'sf' format.

License MIT + file LICENSE

URL https://dieghernan.github.io/nominatimlite/, https://github.com/dieghernan/nominatimlite

BugReports https://github.com/dieghernan/nominatimlite/issues

Depends R (>= 3.6.0)

Imports dplyr (>= 1.0.0), jsonlite (>= 1.7.0), rlang (>= 0.4.9), sf (>= 0.9.0), tibble (>= 3.0.3), utils

Suggests ggplot2 (>= 3.0.0), knitr, osmdata, rmarkdown, testthat (>= 3.0.0), tidygeocoder

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel true

Copyright Data © OpenStreetMap contributors, ODbL 1.0. <https://www.openstreetmap.org/copyright>

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

X-schema.org-applicationCategory cartography

X-schema.org-keywords r, geocoding, openstreetmap, address, nominatim, reverse-geocoding, rstats, shapefile, r-package, spatial, cran, api-wrapper

NeedsCompilation no

Author Diego Hernangómez [aut, cre, cph]
(<https://orcid.org/0000-0001-8457-4658>),
Jindra Lacko [ctb, rev]

Maintainer Diego Hernangómez <diego.hernangomezherrero@gmail.com>

Repository CRAN

Date/Publication 2022-06-10 07:40:02 UTC

R topics documented:

`bbox_to_poly` *Create a bounding box* `sf` *object*

Description

Create a `sf` polygon object from the coordinates of a bounding box

Usage

```
bbox_to_poly(bbox = NA, xmin = NA, ymin = NA, xmax = NA, ymax = NA, crs = 4326)
```

Arguments

`bbox`	numeric vector of 4 elements representing the coordinates of the bounding box. Values should be `c(xmin, ymin, xmax, ymax)`
`xmin, ymin, xmax, ymax`	alternatively, you can use these named parameters instead of `bbox`
`crs`	coordinate reference system, something suitable as input to st_crs

Details

Bounding boxes can be located using different online tools, as Bounding Box Tool.

Value

A sf object

See Also

sf::st_as_sfc()

Other spatial: geo_address_lookup_sf(), geo_amenity_sf(), geo_lite_sf(), reverse_geo_lite_sf()

Other amenity: geo_amenity_sf(), geo_amenity(), osm_amenities

Examples

```
# bounding box of Germany
bbox_GER <- c(5.86631529, 47.27011137, 15.04193189, 55.09916098)

bbox_GER_sf <- bbox_to_poly(bbox_GER)


library(ggplot2)

ggplot(bbox_GER_sf) +
  geom_sf()

# Extract the bounding box of a sf object
Texas <- geo_lite_sf("Texas", points_only = FALSE)
bbox <- sf::st_bbox(Texas)

bbox

bbox_Texas <- bbox_to_poly(bbox)

ggplot(bbox_Texas) +
  geom_sf(col = "red") +
  geom_sf(data = Texas)
```

---

geo_address_lookup *Query the address and other details of one or multiple OSM objects*

Description

Geocodes addresses for OSM objects, identified with the OSM Id.

Usage

```
geo_address_lookup(
  osm_ids,
  type = c("N", "W", "R"),
  lat = "lat",
  long = "lon",
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  custom_query = list()
)
```

Arguments

osm_ids	vector of OSM identifiers (c(00000, 11111, 22222)).
type	vector of the type of the OSM type associated to each osm_ids. Possible values are node ("N"), way ("W") or relation ("R"). If a single value is provided it would be recycled.
lat	latitude column name (i.e. "lat").
long	longitude column name (i.e. "long").
full_results	returns all available data from the geocoding service if TRUE. If FALSE (default) then only latitude and longitude columns are returned from the geocoding service.
return_addresses	return input addresses with results if TRUE. Note that most services return the input addresses with full_results = TRUE and setting return_addresses to FALSE does not prevent this.
verbose	if TRUE then detailed logs are output to the console. FALSE is default. Can be set permanently with options(tidygeocoder.verbose = TRUE)
custom_query	API-specific parameters to be used, passed as a named list (i.e. list(countrycodes = "US")). See Details.

Details

See https://nominatim.org/release-docs/develop/api/Lookup/ for additional parameters to be passed to custom_query.

Value

A tibble with the results.

See Also

Other geocoding: geo_amenity(), geo_lite(), reverse_geo_lite()

Other lookup: geo_address_lookup_sf()

Examples

```
ids <- geo_address_lookup(
  osm_ids = c(46240148, 34633854),
  type = c("W"),
)

ids
```

`geo_address_lookup_sf` *Get spatial objects from OSM ids*

Description

This function allows you to extract the spatial objects for specific OSM objects.

Usage

```
geo_address_lookup_sf(
  osm_ids,
  type = c("N", "W", "R"),
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  custom_query = list(),
  points_only = TRUE
)
```

Arguments

`osm_ids`	vector of OSM identifiers (`c(00000, 11111, 22222)`).
`type`	vector of the type of the OSM type associated to each `osm_ids`. Possible values are node ("N"), way ("W") or relation ("R"). If a single value is provided it would be recycled.
`full_results`	returns all available data from the geocoding service if TRUE. If FALSE (default) then only latitude and longitude columns are returned from the geocoding service.
`return_addresses`	return input addresses with results if TRUE. Note that most services return the input addresses with `full_results = TRUE` and setting return_addresses to FALSE does not prevent this.
`verbose`	if TRUE then detailed logs are output to the console. FALSE is default. Can be set permanently with `options(tidygeocoder.verbose = TRUE)`

custom_query — API-specific parameters to be used, passed as a named list (i.e. `list(countrycodes = "US")`). See Details.

points_only — Logical `TRUE/FALSE`. Whether to return only spatial points (`TRUE`, which is the default) or potentially other shapes as provided by the Nominatim API (`FALSE`).

Details

See https://nominatim.org/release-docs/latest/api/Search/ for additional parameters to be passed to `custom_query`.

Value

A `sf` object with the results.

See Also

Other spatial: `bbox_to_poly()`, `geo_amenity_sf()`, `geo_lite_sf()`, `reverse_geo_lite_sf()`

Other lookup: `geo_address_lookup()`

Examples

```
# Notre Dame Cathedral, Paris

NotreDame <- geo_address_lookup_sf(
  osm_ids = c(201611261),
  type = c("W")
)

library(ggplot2)

ggplot(NotreDame) +
  geom_sf()

NotreDame_poly <- geo_address_lookup_sf(
  osm_ids = c(201611261),
  type = c("W"),
  points_only = FALSE
)

ggplot(NotreDame_poly) +
  geom_sf()
```

`geo_amenity` *Geocode amenities*

Description

This function search amenities as defined by OpenStreetMap on a restricted area defined by a bounding box in the form of (<min_latitude>, <min_longitude>, <max_latitude>, <max_longitude>).

Usage

```
geo_amenity(
  bbox,
  amenity,
  lat = "lat",
  long = "lon",
  limit = 1,
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  custom_query = list(),
  strict = FALSE
)
```

Arguments

`bbox`	A numeric vector of latitude and longitude (<min_latitude>, <min_longitude>, <max_latitude>, <max_longitude>) that restrict the search area. See Details.
`amenity`	A character of a vector of character with the amenities to be geolocated (i.e. `c("pub", "restaurant")`). See Details or osm_amenities.
`lat`	latitude column name (i.e. `"lat"`).
`long`	longitude column name (i.e. `"long"`).
`limit`	maximum number of results to return per input address. Note that each query returns a maximum of 50 results.
`full_results`	returns all available data from the geocoding service if TRUE. If FALSE (default) then only latitude and longitude columns are returned from the geocoding service.
`return_addresses`	return input addresses with results if TRUE. Note that most services return the input addresses with `full_results = TRUE` and setting return_addresses to FALSE does not prevent this.
`verbose`	if TRUE then detailed logs are output to the console. FALSE is default. Can be set permanently with `options(tidygeocoder.verbose = TRUE)`
`custom_query`	API-specific parameters to be used. See `geo_lite()`.
`strict`	Logical TRUE/FALSE. Force the results to be included inside the bbox. Note that Nominatim default behaviour may return results located outside the provided bounding box.

Details

Bounding boxes can be located using different online tools, as Bounding Box Tool.

For a full list of valid amenities see https://wiki.openstreetmap.org/wiki/Key:amenity.

Value

A tibble with the results.

See Also

Other amenity: bbox_to_poly(), geo_amenity_sf(), osm_amenities

Other geocoding: geo_address_lookup(), geo_lite(), reverse_geo_lite()

Examples

```
# Times Square, NY, USA
bbox <- c(
  -73.9894467311, 40.75573629,
  -73.9830630737, 40.75789245
)

geo_amenity(
  bbox = bbox,
  amenity = "restaurant"
)

# Several amenities
geo_amenity(
  bbox = bbox,
  amenity = c("restaurant", "pub")
)

# Increase limit and use with strict
geo_amenity(
  bbox = bbox,
  amenity = c("restaurant", "pub"),
  limit = 10,
  strict = TRUE
)
```

geo_amenity_sf *Get spatial objects of amenities*

Description

This function search amenities as defined by OpenStreetMap on a restricted area defined by a bounding box in the form of (<min_latitude>, <min_longitude>, <max_latitude>, <max_longitude>).

Usage

```
geo_amenity_sf(
  bbox,
  amenity,
  limit = 1,
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  custom_query = list(),
  points_only = TRUE,
  strict = FALSE
)
```

Arguments

bbox	A numeric vector of latitude and longitude (<min_latitude>, <min_longitude>, <max_latitude>, <max_longitude>) that restrict the search area. See Details.
amenity	A character of a vector of character with the amenities to be geolocated (i.e. `c("pub", "restaurant")`). See Details or osm_amenities.
limit	maximum number of results to return per input address. Note that each query returns a maximum of 50 results.
full_results	returns all available data from the geocoding service if TRUE. If FALSE (default) then only latitude and longitude columns are returned from the geocoding service.
return_addresses	return input addresses with results if TRUE. Note that most services return the input addresses with `full_results = TRUE` and setting return_addresses to FALSE does not prevent this.
verbose	if TRUE then detailed logs are output to the console. FALSE is default. Can be set permanently with `options(tidygeocoder.verbose = TRUE)`
custom_query	API-specific parameters to be used. See `geo_lite()`.
points_only	Logical `TRUE/FALSE`. Whether to return only spatial points (`TRUE`, which is the default) or potentially other shapes as provided by the Nominatim API (`FALSE`).
strict	Logical TRUE/FALSE. Force the results to be included inside the bbox. Note that Nominatim default behaviour may return results located outside the provided bounding box.

Details

Bounding boxes can be located using different online tools, as Bounding Box Tool.

For a full list of valid amenities see https://wiki.openstreetmap.org/wiki/Key:amenity.

Value

A sf object with the results.

See Also

Other spatial: bbox_to_poly(), geo_address_lookup_sf(), geo_lite_sf(), reverse_geo_lite_sf()

Other amenity: bbox_to_poly(), geo_amenity(), osm_amenities

Examples

```
# Madrid, Spain

library(ggplot2)

bbox <- c(
  -3.888954, 40.311977,
  -3.517916, 40.643729
)

# Restaurants and pubs

rest_pub <- geo_amenity_sf(bbox,
  c("restaurant", "pub"),
  limit = 50
)


ggplot(rest_pub) +
  geom_sf()

# Hospital as polygon

hosp <- geo_amenity_sf(bbox,
  "hospital",
  points_only = FALSE
)

ggplot(hosp) +
  geom_sf()
```

`geo_lite` *Geocode addresses*

Description

Geocodes addresses given as character values.

Usage

```
geo_lite(
  address,
  lat = "lat",
  long = "lon",
  limit = 1,
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  custom_query = list()
)
```

Arguments

`address`	single line address (i.e. `"1600 Pennsylvania Ave NW, Washington"`) or a vector of addresses (`c("Madrid", "Barcelona")`).
`lat`	latitude column name (i.e. `"lat"`).
`long`	longitude column name (i.e. `"long"`).
`limit`	maximum number of results to return per input address. Note that each query returns a maximum of 50 results.
`full_results`	returns all available data from the geocoding service if TRUE. If FALSE (default) then only latitude and longitude columns are returned from the geocoding service.
`return_addresses`	return input addresses with results if TRUE. Note that most services return the input addresses with `full_results = TRUE` and setting return_addresses to FALSE does not prevent this.
`verbose`	if TRUE then detailed logs are output to the console. FALSE is default. Can be set permanently with `options(tidygeocoder.verbose = TRUE)`
`custom_query`	API-specific parameters to be used, passed as a named list (i.e. `list(countrycodes = "US")`). See Details.

Details

See https://nominatim.org/release-docs/latest/api/Search/ for additional parameters to be passed to `custom_query`.

Value

A `tibble` with the results.

See Also

`geo_lite_sf()`, `tidygeocoder::geo()`

Other geocoding: `geo_address_lookup()`, `geo_amenity()`, `reverse_geo_lite()`

Examples

```
geo_lite("Madrid, Spain")

# Several addresses
geo_lite(c("Madrid", "Barcelona"))

# With options: restrict search to USA
geo_lite(c("Madrid", "Barcelona"),
  custom_query = list(countrycodes = "US"),
  full_results = TRUE
)
```

`geo_lite_sf` *Get spatial objects through geocoding*

Description

This function allows you to geocode addresses and return the corresponding spatial object.

The parameter `points_only` specifies whether the function results will be points (all Nominatim results are guaranteed to have at least point geometry) or possibly other spatial objects.

Note that the type of geometry returned in case of `points_only = FALSE` will depend on the object being geocoded:

- administrative areas, major buildings and the like will be returned as polygons
- rivers, roads and their like as lines
- amenities may be points even in case of a `points_only = FALSE` call

The function is vectorized, allowing for multiple addresses to be geocoded; in case of `points_only = FALSE` multiple geometry types may be returned.

Usage

```
geo_lite_sf(
  address,
  limit = 1,
  return_addresses = TRUE,
  full_results = FALSE,
  verbose = FALSE,
  custom_query = list(),
  points_only = TRUE
)
```

Arguments

address	single line address (i.e. "1600 Pennsylvania Ave NW, Washington") or a vector of addresses (c("Madrid", "Barcelona")).
limit	maximum number of results to return per input address. Note that each query returns a maximum of 50 results.
return_addresses	return input addresses with results if TRUE. Note that most services return the input addresses with full_results = TRUE and setting return_addresses to FALSE does not prevent this.
full_results	returns all available data from the geocoding service if TRUE. If FALSE (default) then only latitude and longitude columns are returned from the geocoding service.
verbose	if TRUE then detailed logs are output to the console. FALSE is default. Can be set permanently with options(tidygeocoder.verbose = TRUE)
custom_query	API-specific parameters to be used, passed as a named list (i.e. list(countrycodes = "US")). See Details.
points_only	Logical TRUE/FALSE. Whether to return only spatial points (TRUE, which is the default) or potentially other shapes as provided by the Nominatim API (FALSE).

Details

See https://nominatim.org/release-docs/latest/api/Search/ for additional parameters to be passed to custom_query.

Value

A sf object with the results.

See Also

geo_lite()

Other spatial: bbox_to_poly(), geo_address_lookup_sf(), geo_amenity_sf(), reverse_geo_lite_sf()

Examples

```
# Map - Points
library(ggplot2)
pentagon <- geo_lite_sf("Pentagon")


ggplot(pentagon) +
  geom_sf()

pentagon_poly <- geo_lite_sf("Pentagon", points_only = FALSE)

ggplot(pentagon_poly) +
  geom_sf()

# Several results

Madrid <- geo_lite_sf("Madrid",
  limit = 2,
  points_only = FALSE, full_results = TRUE
)


ggplot(Madrid) +
  geom_sf(fill = NA)

Starbucks <- geo_lite_sf("Starbucks, New York",
  limit = 20, full_results = TRUE
)


ggplot(Starbucks) +
  geom_sf()
```

`osm_amenities` *OpenStreetMap amenity database*

Description

Database with the list of amenities available on OpenStreetMap.

Format

A `tibble` with the amenities and the corresponding category

Details

category	amenity
Sustenance	bar
Sustenance	biergarten
Sustenance	cafe
Sustenance	fast_food
Sustenance	food_court
Sustenance	ice_cream
Sustenance	pub
Sustenance	restaurant
Education	college
Education	driving_school
Education	kindergarten
Education	language_school
Education	library
Education	toy_library
Education	music_school
Education	school
Education	university
Transportation	bicycle_parking
Transportation	bicycle_repair_station
Transportation	bicycle_rental
Transportation	boat_rental
Transportation	boat_sharing
Transportation	bus_station
Transportation	car_rental
Transportation	car_sharing
Transportation	car_wash
Transportation	vehicle_inspection
Transportation	charging_station
Transportation	ferry_terminal
Transportation	fuel
Transportation	grit_bin
Transportation	motorcycle_parking
Transportation	parking
Transportation	parking_entrance
Transportation	parking_space
Transportation	taxi
Financial	atm
Financial	bank
Financial	bureau_de_change
Healthcare	baby_hatch
Healthcare	clinic
Healthcare	dentist
Healthcare	doctors
Healthcare	hospital
Healthcare	nursing_home
Healthcare	pharmacy
Healthcare	social_facility

Healthcare	veterinary
Entertainment-Arts-Culture	arts_centre
Entertainment-Arts-Culture	brothel
Entertainment-Arts-Culture	casino
Entertainment-Arts-Culture	cinema
Entertainment-Arts-Culture	community_centre
Entertainment-Arts-Culture	conference_centre
Entertainment-Arts-Culture	events_venue
Entertainment-Arts-Culture	fountain
Entertainment-Arts-Culture	gambling
Entertainment-Arts-Culture	love_hotel
Entertainment-Arts-Culture	nightclub
Entertainment-Arts-Culture	planetarium
Entertainment-Arts-Culture	public_bookcase
Entertainment-Arts-Culture	social_centre
Entertainment-Arts-Culture	stripclub
Entertainment-Arts-Culture	studio
Entertainment-Arts-Culture	swingerclub
Entertainment-Arts-Culture	theatre
Public Service	courthouse
Public Service	embassy
Public Service	fire_station
Public Service	police
Public Service	post_box
Public Service	post_depot
Public Service	post_office
Public Service	prison
Public Service	ranger_station
Public Service	townhall
Facilities	bbq
Facilities	bench
Facilities	dog_toilet
Facilities	drinking_water
Facilities	give_box
Facilities	shelter
Facilities	shower
Facilities	telephone
Facilities	toilets
Facilities	water_point
Facilities	watering_place
Waste Management	sanitary_dump_station
Waste Management	recycling
Waste Management	waste_basket
Waste Management	waste_disposal
Waste Management	waste_transfer_station
Others	animal_boarding
Others	animal_breeding
Others	animal_shelter

Others	baking_oven
Others	childcare
Others	clock
Others	crematorium
Others	dive_centre

Note

Data extracted on **14 June 2021**.

Source

https://wiki.openstreetmap.org/wiki/Key:amenity

See Also

Other amenity: `bbox_to_poly()`, `geo_amenity_sf()`, `geo_amenity()`

Examples

```
amenities <- nominatimlite::osm_amenities

amenities
```

---

reverse_geo_lite *Reverse geocode coordinates*

Description

Reverse geocodes geographic coordinates (latitude and longitude) given as numeric values. Latitudes must be between -90 and 90 and longitudes must be between -180 and 180.

Usage

```
reverse_geo_lite(
  lat,
  long,
  address = "address",
  full_results = FALSE,
  return_coords = TRUE,
  verbose = FALSE,
  custom_query = list()
)
```

Arguments

`lat`	latitude values (input data)
`long`	longitude values (input data)
`address`	name of the address column (in the output data)
`full_results`	returns all available data from the geocoding service if TRUE. If FALSE (default) then only a single address column is returned from the geocoding service.
`return_coords`	return input coordinates with results if TRUE. Note that most services return the input coordinates with `full_results = TRUE` and setting `return_coords` to FALSE does not prevent this.
`verbose`	if TRUE then detailed logs are output to the console. FALSE is default. Can be set permanently with `options(tidygeocoder.verbose = TRUE)`
`custom_query`	API-specific parameters to be used, passed as a named list (ie. `list(zoom = 3)`). See Details.

Details

See https://nominatim.org/release-docs/develop/api/Reverse/ for additional parameters to be passed to `custom_query`.

Use the option `custom_query = list(zoom = 3)` to adjust the output. Some equivalences on terms of zoom:

zoom	address_detail
3	country
5	state
8	county
10	city
14	suburb
16	major streets
17	major and minor streets
18	building

Value

A `tibble` with the results.

See Also

`reverse_geo_lite_sf()`, `tidygeocoder::reverse_geo()`

Other geocoding: `geo_address_lookup()`, `geo_amenity()`, `geo_lite()`

Examples

```
reverse_geo_lite(lat = 40.75728, long = -73.98586)

# Several coordinates
reverse_geo_lite(
  lat = c(40.75728, 55.95335),
  long = c(-73.98586, -3.188375)
)

# With options: zoom to country
reverse_geo_lite(
  lat = c(40.75728, 55.95335),
  long = c(-73.98586, -3.188375),
  custom_query = list(zoom = 0),
  verbose = TRUE,
  full_results = TRUE
)
```

`reverse_geo_lite_sf` *Get spatial objects through reverse geocoding*

Description

This function allows you extract the spatial object located on a known pair of coordinates (lat, long). Latitudes must be between -90 and 90 and longitudes must be between -180 and 180.

Usage

```
reverse_geo_lite_sf(
  lat,
  long,
  address = "address",
  full_results = FALSE,
  return_coords = TRUE,
  verbose = FALSE,
  custom_query = list(),
  points_only = TRUE
)
```

Arguments

`lat`	latitude values (input data)
`long`	longitude values (input data)
`address`	name of the address column (in the output data)
`full_results`	returns all available data from the geocoding service if TRUE. If FALSE (default) then only a single address column is returned from the geocoding service.

return_coords	return input coordinates with results if TRUE. Note that most services return the input coordinates with full_results = TRUE and setting return_coords to FALSE does not prevent this.
verbose	if TRUE then detailed logs are output to the console. FALSE is default. Can be set permanently with options(tidygeocoder.verbose = TRUE)
custom_query	API-specific parameters to be used, passed as a named list (ie. list(zoom = 3)). See Details.
points_only	Logical TRUE/FALSE. Whether to return only spatial points (TRUE, which is the default) or potentially other shapes as provided by the Nominatim API (FALSE).

Details

See https://nominatim.org/release-docs/develop/api/Reverse/ for additional parameters to be passed to custom_query.

Use the option custom_query = list(zoom = 3) to adjust the output. Some equivalences on terms of zoom:

zoom	address_detail
3	country
5	state
8	county
10	city
14	suburb
16	major streets
17	major and minor streets
18	building

Value

A sf object with the results.

See Also

reverse_geo_lite()

Other spatial: bbox_to_poly(), geo_address_lookup_sf(), geo_amenity_sf(), geo_lite_sf()

Examples

```
library(ggplot2)


Coliseum <- geo_lite("Coliseo, Rome, Italy")

# Coliseum
Col_sf <- reverse_geo_lite_sf(
  lat = Coliseum$lat,
```

```
  lon = Coliseum$lon,
  points_only = FALSE
)

ggplot(Col_sf) +
  geom_sf()

# City of Rome - Zoom 10

Rome_sf <- reverse_geo_lite_sf(
  lat = Coliseum$lat,
  lon = Coliseum$lon,
  custom_query = list(zoom = 10),
  points_only = FALSE
)

ggplot(Rome_sf) +
  geom_sf()

# County - Zoom 8

County_sf <- reverse_geo_lite_sf(
  lat = Coliseum$lat,
  lon = Coliseum$lon,
  custom_query = list(zoom = 8),
  points_only = FALSE
)

ggplot(County_sf) +
  geom_sf()
```

Index