Package 'nominatimlite'

August 25, 2026

Type Package

Title Interface to the 'Nominatim' API

Version 0.7.0

Description Provides a lightweight interface to the 'Nominatim' API <https://nominatim.org/release-docs/latest/>. It supports free-form and structured address searches, searches for addresses from coordinates, amenity lookup and address lookup by 'OpenStreetMap' object identifier. It returns results as 'tibble' data frames or 'sf' objects.

License MIT + file LICENSE

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

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

Depends R (>= 4.1.0)

Imports dplyr (>= 1.0.0), jsonlite (>= 1.7.0), sf (>= 0.9.0), tools, utils

Suggests curl, ggplot2, knitr, quarto, testthat (>= 3.3.0), tibble, withr

VignetteBuilder quarto

Config/Needs/website dieghernan/gitdevr, pak, tidyverse, leaflet, reactable, crosstalk, tidyr, htmltools, arcgeocoder, tidygeocoder

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Config/testthat/parallel false

Copyright See file inst/COPYRIGHTS.

Encoding UTF-8

LazyData true

X-schema.org-applicationCategory cartography

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

NeedsCompilation no

Author Diego Hernangómez [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-8457-4658>), Jindra Lacko [ctb, rev] (ORCID: <https://orcid.org/0000-0002-0375-5156>), Alex White [ctb], OpenStreetMap contributors [cph] (For the osm_amenities data)

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

Repository CRAN

Date/Publication 2026-08-25 11:20:18 UTC

Contents

bbox_to_poly 2
geo_address_lookup 4
geo_address_lookup_sf 5
geo_amenity 7
geo_amenity_sf 10
geo_lite 12
geo_lite_sf 14
geo_lite_struct 16
geo_lite_struct_sf 18
osm_amenities 20
reverse_geo_lite 21
reverse_geo_lite_sf 23

Index **26**

`bbox_to_poly` *Convert a bounding box to an* `sfc` `POLYGON` *object*

Description

Converts bounding box coordinates to an `sfc` object with `POLYGON` geometry.

Usage

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

Arguments

bbox	A numeric vector of four bounding box coordinates in the form c(xmin, ymin, xmax, ymax).
xmin, ymin, xmax, ymax	A numeric value specifying an individual bounding box coordinate. Use these arguments as an alternative to bbox.
crs	coordinate reference system, something suitable as input to st_crs

Details

Bounding boxes can be located using online tools such as https://boundingbox.klokantech.com/.

Value

An sfc object with POLYGON geometry and the coordinate reference system specified by crs.

See Also

sf::st_as_sfc() and sf::st_sfc().

Spatial helper and output functions: geo_address_lookup_sf(), geo_amenity_sf(), geo_lite_sf(), geo_lite_struct_sf(), reverse_geo_lite_sf()

Examples

```
# Convert the bounding box for 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 an `sf` object.
sfobj <- geo_lite_sf("seychelles", points_only = FALSE)

sfobj

# Require at least one non-empty object.
if (!all(sf::st_is_empty(sfobj))) {
  bbox <- sf::st_bbox(sfobj)

  bbox

  bbox_sfobj <- bbox_to_poly(bbox)

  ggplot(bbox_sfobj) +
    geom_sf(fill = "lightblue", alpha = 0.5) +
```

```
    geom_sf(data = sfobj, fill = "wheat")
}
```

`geo_address_lookup` *Look up OpenStreetMap objects*

Description

Looks up addresses and other details for one or more OpenStreetMap (OSM) objects, such as nodes, ways or relations. Results are returned as a tibble. Use `geo_address_lookup_sf()` to return an `sf` object instead.

Usage

```
geo_address_lookup(
  osm_ids,
  type = c("N", "W", "R"),
  lat = "lat",
  long = "lon",
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  nominatim_server = "https://nominatim.openstreetmap.org/",
  custom_query = list()
)
```

Arguments

`osm_ids`	A numeric vector of OSM identifiers, for example `c(12345, 67890)`.
`type`	A character vector containing the OSM object type associated with each value in `osm_ids`. Possible values are node (`"N"`), way (`"W"`) and relation (`"R"`). A single value is recycled.
`lat`	A character string specifying the name of the latitude column in the output. Defaults to `"lat"`.
`long`	A character string specifying the name of the longitude column in the output. Defaults to `"lon"`.
`full_results`	A logical value indicating whether to return all available fields from the Nominatim API. If `FALSE`, only query metadata, location data and requested address columns are returned.
`return_addresses`	A logical value indicating whether to include single-line addresses in the results.
`verbose`	A logical value indicating whether to display detailed messages in the console.

`nominatim_server`
: A character string specifying the base URL of the Nominatim server. Defaults to `"https://nominatim.openstreetmap.org/"`.

`custom_query`
: A named list of additional API parameters, for example `list(countrycodes = "US")`. See **Details**.

Details

See https://nominatim.org/release-docs/latest/api/Lookup/ for additional parameters to pass to `custom_query`.

Value

A `tibble` with the results that match the query.

See Also

`geo_lite()` for finding objects when their OSM identifiers are unknown.

Address lookup functions: `geo_address_lookup_sf()`

Examples

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

ids

several <- geo_address_lookup(c(146656, 240109189), type = c("R", "N"))
several
```

---

geo_address_lookup_sf *Look up OpenStreetMap objects and return* R*hrefhttps://CRAN.R-project.org/package=sf***sf** *objects*

Description

Looks up addresses and other details for one or more OpenStreetMap (OSM) objects, such as nodes, ways or relations. Results are returned as an `sf` object. Use `geo_address_lookup()` to return a tibble instead.

Usage

```
geo_address_lookup_sf(
  osm_ids,
  type = c("N", "W", "R"),
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  nominatim_server = "https://nominatim.openstreetmap.org/",
  custom_query = list(),
  points_only = TRUE
)
```

Arguments

`osm_ids`	A numeric vector of OSM identifiers, for example `c(12345, 67890)`.
`type`	A character vector containing the OSM object type associated with each value in `osm_ids`. Possible values are node (`"N"`), way (`"W"`) and relation (`"R"`). A single value is recycled.
`full_results`	A logical value indicating whether to return all available fields from the Nominatim API. If `FALSE`, only query metadata, geometry and requested address columns are returned.
`return_addresses`	A logical value indicating whether to include single-line addresses in the results.
`verbose`	A logical value indicating whether to display detailed messages in the console.
`nominatim_server`	A character string specifying the base URL of the Nominatim server. Defaults to `"https://nominatim.openstreetmap.org/"`.
`custom_query`	A named list of additional API parameters, for example `list(countrycodes = "US")`. See **Details**.
`points_only`	A logical value indicating whether to return only point geometries. If `FALSE`, the API may return other geometry types. See **About geometry types**.

Details

See https://nominatim.org/release-docs/latest/api/Lookup/ for additional parameters to pass to `custom_query`.

Value

An `sf` object with the results that match the query.

About geometry types

The `points_only` argument controls whether the results contain only points. All Nominatim results have at least a point geometry.

When `points_only = FALSE`, the geometry type depends on the matching feature. Administrative areas and major buildings are returned as polygons, rivers and roads are returned as lines and amenities may still be returned as points.

This function is vectorized, allowing multiple addresses to be searched. With `points_only = FALSE`, multiple geometry types may be returned.

See Also

Address lookup functions: `geo_address_lookup()`

Spatial helper and output functions: `bbox_to_poly()`, `geo_amenity_sf()`, `geo_lite_sf()`, `geo_lite_struct_sf()`, `reverse_geo_lite_sf()`

Examples

```
# Look up Notre-Dame Cathedral in Paris.

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

# Require at least one non-empty object.
if (!all(sf::st_is_empty(NotreDame))) {
  library(ggplot2)

  ggplot(NotreDame) +
    geom_sf()
}

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

if (!all(sf::st_is_empty(NotreDame_poly))) {
  ggplot(NotreDame_poly) +
    geom_sf()
}

# Look up multiple OSM objects.

several <- geo_address_lookup_sf(c(146656, 240109189), type = c("R", "N"))
several
```

`geo_amenity` *Look up OpenStreetMap amenities*

Description

Looks up OpenStreetMap amenities within a bounding box of the form `(xmin, ymin, xmax, ymax)`. Results are returned as a tibble. Use `geo_amenity_sf()` to return an `sf` object instead.

Usage

```
geo_amenity(
  bbox,
  amenity,
  lat = "lat",
  long = "lon",
  limit = 1,
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  nominatim_server = "https://nominatim.openstreetmap.org/",
  progressbar = TRUE,
  custom_query = list(),
  strict = FALSE
)
```

Arguments

`bbox`	A numeric vector, an `sf` object or an `sfc` object specifying a bounding box (viewbox) used to limit the search. Numeric vectors must contain **longitude** (`x`) and **latitude** (`y`) in the form `(xmin, ymin, xmax, ymax)`. See **Details**.
`amenity`	A character vector of amenities to look up, for example `c("pub", "restaurant")`. See osm_amenities.
`lat`	A character string specifying the name of the latitude column in the output. Defaults to `"lat"`.
`long`	A character string specifying the name of the longitude column in the output. Defaults to `"lon"`.
`limit`	A positive integer specifying the maximum number of results to return per query. Nominatim returns at most 50 results per query.
`full_results`	A logical value indicating whether to return all available fields from the Nominatim API. If `FALSE`, only query metadata, location data and requested address columns are returned.
`return_addresses`	A logical value indicating whether to include single-line addresses in the results.
`verbose`	A logical value indicating whether to display detailed messages in the console.
`nominatim_server`	A character string specifying the base URL of the Nominatim server. Defaults to `"https://nominatim.openstreetmap.org/"`.
`progressbar`	A logical value indicating whether to display a progress bar when processing multiple queries.

custom_query — A named list of additional API parameters, for example `list(countrycodes = "US")`. See **Details**.

strict — A logical value indicating whether to keep only results inside bbox. If `FALSE` (the default), Nominatim may return results outside the bounding box.

Details

Bounding boxes can be located using online tools such as https://boundingbox.klokantech.com/.

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

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

Value

A `tibble` with the results that match the query.

See Also

`bbox_to_poly()` for converting bounding box coordinates to a **sf** polygon.

Amenity lookup functions and data: `geo_amenity_sf()`, `osm_amenities`

Examples

```
# Define a bounding box around Times Square, New York.
bbox <- c(
  -73.9894467311, 40.75573629,
  -73.9830630737, 40.75789245
)

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

# Search for multiple amenities.
geo_amenity(
  bbox = bbox,
  amenity = c("restaurant", "pub")
)

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

geo_amenity_sf *Look up OpenStreetMap amenities and return* R*hrefhttps://CRAN.R-project.org/package=sf***sf** *objects*

Description

Looks up OpenStreetMap amenities within a bounding box of the form `(xmin, ymin, xmax, ymax)`. Results are returned as an `sf` object. Use `geo_amenity()` to return a tibble instead.

Usage

```
geo_amenity_sf(
  bbox,
  amenity,
  limit = 1,
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  nominatim_server = "https://nominatim.openstreetmap.org/",
  progressbar = TRUE,
  custom_query = list(),
  strict = FALSE,
  points_only = TRUE
)
```

Arguments

`bbox`	A numeric vector, an `sf` object or an `sfc` object specifying a bounding box (viewbox) used to limit the search. Numeric vectors must contain **longitude** (`x`) and **latitude** (`y`) in the form `(xmin, ymin, xmax, ymax)`. See **Details**.
`amenity`	A character vector of amenities to look up, for example `c("pub", "restaurant")`. See osm_amenities.
`limit`	A positive integer specifying the maximum number of results to return per query. Nominatim returns at most 50 results per query.
`full_results`	A logical value indicating whether to return all available fields from the Nominatim API. If `FALSE`, only query metadata, geometry and requested address columns are returned.
`return_addresses`	A logical value indicating whether to include single-line addresses in the results.
`verbose`	A logical value indicating whether to display detailed messages in the console.
`nominatim_server`	A character string specifying the base URL of the Nominatim server. Defaults to `"https://nominatim.openstreetmap.org/"`.

progressbar	A logical value indicating whether to display a progress bar when processing multiple queries.
custom_query	A named list of additional API parameters, for example list(countrycodes = "US"). See **Details**.
strict	A logical value indicating whether to keep only results inside bbox. If FALSE (the default), Nominatim may return results outside the bounding box.
points_only	A logical value indicating whether to return only point geometries. If FALSE, the API may return other geometry types. See **About geometry types**.

Details

Bounding boxes can be located using online tools such as https://boundingbox.klokantech.com/.

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

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

Value

An sf object with the results that match the query.

About geometry types

The points_only argument controls whether the results contain only points. All Nominatim results have at least a point geometry.

When points_only = FALSE, the geometry type depends on the matching feature. Administrative areas and major buildings are returned as polygons, rivers and roads are returned as lines and amenities may still be returned as points.

This function is vectorized, allowing multiple addresses to be searched. With points_only = FALSE, multiple geometry types may be returned.

See Also

Amenity lookup functions and data: geo_amenity(), osm_amenities

Spatial helper and output functions: bbox_to_poly(), geo_address_lookup_sf(), geo_lite_sf(), geo_lite_struct_sf(), reverse_geo_lite_sf()

Examples

```
# Retrieve the Usera district in Madrid.

library(ggplot2)
mad <- geo_lite_sf("Usera, Madrid, Spain", points_only = FALSE)

# Search for restaurants, pubs and schools.
```

```
rest_pub <- geo_amenity_sf(mad, c("restaurant", "pub", "school"),
  limit = 50
)

if (!all(sf::st_is_empty(rest_pub))) {
  ggplot(mad) +
    geom_sf() +
    geom_sf(data = rest_pub, aes(color = query, shape = query))
}
```

`geo_lite` *Search for addresses with free-form queries*

Description

Searches for addresses supplied as a character vector and returns matching results as a tibble. Use `geo_lite_sf()` to return an `sf` object instead.

This function performs the **free-form address search** described in the API endpoint.

Usage

```
geo_lite(
  address,
  lat = "lat",
  long = "lon",
  limit = 1,
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  nominatim_server = "https://nominatim.openstreetmap.org/",
  progressbar = TRUE,
  custom_query = list()
)
```

Arguments

`address`	A character vector of single-line addresses, for example `"1600 Pennsylvania Ave NW, Washington"` or `c("Madrid", "Barcelona")`.
`lat`	A character string specifying the name of the latitude column in the output. Defaults to `"lat"`.
`long`	A character string specifying the name of the longitude column in the output. Defaults to `"lon"`.
`limit`	A positive integer specifying the maximum number of results to return per query. Nominatim returns at most 50 results per query.

full_results A logical value indicating whether to return all available fields from the Nominatim API. If FALSE, only query metadata, location data and requested address columns are returned.

return_addresses A logical value indicating whether to include single-line addresses in the results.

verbose A logical value indicating whether to display detailed messages in the console.

nominatim_server A character string specifying the base URL of the Nominatim server. Defaults to "https://nominatim.openstreetmap.org/".

progressbar A logical value indicating whether to display a progress bar when processing multiple queries.

custom_query A named list of additional API parameters, for example list(countrycodes = "US"). See **Details**.

Details

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

Value

A tibble with the results that match the query.

See Also

reverse_geo_lite() for reverse geocoding coordinates, geo_address_lookup() for looking up known OpenStreetMap identifiers and geo_amenity() for searching amenities within a bounding box.

Address search functions: geo_lite_sf(), geo_lite_struct(), geo_lite_struct_sf()

Examples

```
geo_lite("Madrid, Spain")

# Search for multiple addresses.
geo_lite(c("Madrid", "Barcelona"))

# Restrict the search to the United States and return all fields.
geo_lite(c("Madrid", "Barcelona"),
  custom_query = list(countrycodes = "US"),
  full_results = TRUE
)
```

geo_lite_sf *Search for addresses with free-form queries and return* R*hrefhttps://CRAN.R-project.org/package=sf*sf *objects*

Description

Searches for addresses supplied as a character vector and returns matching results as an sf object. Use geo_lite() to return a tibble instead.

This function performs the **free-form address search** described in the API endpoint.

Usage

```
geo_lite_sf(
  address,
  limit = 1,
  return_addresses = TRUE,
  full_results = FALSE,
  verbose = FALSE,
  progressbar = TRUE,
  nominatim_server = "https://nominatim.openstreetmap.org/",
  custom_query = list(),
  points_only = TRUE
)
```

Arguments

address	A character vector of single-line addresses, for example "1600 Pennsylvania Ave NW, Washington" or c("Madrid", "Barcelona").
limit	A positive integer specifying the maximum number of results to return per query. Nominatim returns at most 50 results per query.
return_addresses	A logical value indicating whether to include single-line addresses in the results.
full_results	A logical value indicating whether to return all available fields from the Nominatim API. If FALSE, only query metadata, geometry and requested address columns are returned.
verbose	A logical value indicating whether to display detailed messages in the console.
progressbar	A logical value indicating whether to display a progress bar when processing multiple queries.
nominatim_server	A character string specifying the base URL of the Nominatim server. Defaults to "https://nominatim.openstreetmap.org/".
custom_query	A named list of additional API parameters, for example list(countrycodes = "US"). See **Details**.
points_only	A logical value indicating whether to return only point geometries. If FALSE, the API may return other geometry types. See **About geometry types**.

Details

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

Value

An sf object with the results that match the query.

About geometry types

The points_only argument controls whether the results contain only points. All Nominatim results have at least a point geometry.

When points_only = FALSE, the geometry type depends on the matching feature. Administrative areas and major buildings are returned as polygons, rivers and roads are returned as lines and amenities may still be returned as points.

This function is vectorized, allowing multiple addresses to be searched. With points_only = FALSE, multiple geometry types may be returned.

See Also

Address search functions: geo_lite(), geo_lite_struct(), geo_lite_struct_sf()

Spatial helper and output functions: bbox_to_poly(), geo_address_lookup_sf(), geo_amenity_sf(), geo_lite_struct_sf(), reverse_geo_lite_sf()

Examples

```
# Return point geometries.
library(ggplot2)

string <- "Statue of Liberty, NY, USA"
sol <- geo_lite_sf(string)

if (!all(sf::st_is_empty(sol))) {
  ggplot(sol) +
    geom_sf()
}

sol_poly <- geo_lite_sf(string, points_only = FALSE)

if (!all(sf::st_is_empty(sol_poly))) {
  ggplot(sol_poly) +
    geom_sf() +
    geom_sf(data = sol, color = "red")
}
# Return multiple matches.

madrid <- geo_lite_sf("Comunidad de Madrid, Spain",
  limit = 2,
  points_only = FALSE, full_results = TRUE
```

```
)

if (!all(sf::st_is_empty(madrid))) {
  ggplot(madrid) +
    geom_sf(fill = NA)
}
```

`geo_lite_struct` *Search for addresses with structured queries*

Description

Searches for addresses already split into components and returns matching results as a tibble. Use `geo_lite_struct_sf()` to return an `sf` object instead.

This function performs the **structured address search** described in the API endpoint. To perform a free-form search, use `geo_lite()`.

Usage

```
geo_lite_struct(
  amenity = NULL,
  street = NULL,
  city = NULL,
  county = NULL,
  state = NULL,
  country = NULL,
  postalcode = NULL,
  lat = "lat",
  long = "lon",
  limit = 1,
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  nominatim_server = "https://nominatim.openstreetmap.org/",
  custom_query = list()
)
```

Arguments

`amenity`	A character string specifying the name or type of amenity. See `geo_amenity()`.
`street`	A character string specifying the house number and street name.
`city`	A character string specifying the city.
`county`	A character string specifying the county.
`state`	A character string specifying the state.

country — A character string specifying the country.

postalcode — A character string specifying the postal code.

lat — A character string specifying the name of the latitude column in the output. Defaults to "lat".

long — A character string specifying the name of the longitude column in the output. Defaults to "lon".

limit — A positive integer specifying the maximum number of results to return per query. Nominatim returns at most 50 results per query.

full_results — A logical value indicating whether to return all available fields from the Nominatim API. If FALSE, only query metadata, location data and requested address columns are returned.

return_addresses — A logical value indicating whether to include single-line addresses in the results.

verbose — A logical value indicating whether to display detailed messages in the console.

nominatim_server — A character string specifying the base URL of the Nominatim server. Defaults to "https://nominatim.openstreetmap.org/".

custom_query — A named list of additional API parameters, for example list(countrycodes = "US"). See **Details**.

Details

A structured address search accepts an address already split into components. Each argument represents an address field. All components are optional, so provide only those relevant to the address you want to find.

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

Value

A tibble with the results that match the query.

See Also

Address search functions: geo_lite(), geo_lite_sf(), geo_lite_struct_sf()

Examples

```
pl_mayor <- geo_lite_struct(
  street = "Plaza Mayor", country = "Spain",
  limit = 50, full_results = TRUE
)

dplyr::glimpse(pl_mayor)
```

geo_lite_struct_sf *Search for addresses with structured queries and return* R*hrefhttps://CRAN.R-project.org/package=sf***sf** *objects*

Description

Searches for addresses already split into components and returns matching results as an sf object. Use geo_lite_struct() to return a tibble instead.

This function performs the **structured address search** described in the API endpoint. To perform a free-form search, use geo_lite_sf().

Usage

```
geo_lite_struct_sf(
  amenity = NULL,
  street = NULL,
  city = NULL,
  county = NULL,
  state = NULL,
  country = NULL,
  postalcode = NULL,
  limit = 1,
  full_results = FALSE,
  return_addresses = TRUE,
  verbose = FALSE,
  nominatim_server = "https://nominatim.openstreetmap.org/",
  custom_query = list(),
  points_only = TRUE
)
```

Arguments

amenity	A character string specifying the name or type of amenity. See geo_amenity().
street	A character string specifying the house number and street name.
city	A character string specifying the city.
county	A character string specifying the county.
state	A character string specifying the state.
country	A character string specifying the country.
postalcode	A character string specifying the postal code.
limit	A positive integer specifying the maximum number of results to return per query. Nominatim returns at most 50 results per query.
full_results	A logical value indicating whether to return all available fields from the Nominatim API. If FALSE, only query metadata, geometry and requested address columns are returned.

`return_addresses`
: A logical value indicating whether to include single-line addresses in the results.

`verbose`
: A logical value indicating whether to display detailed messages in the console.

`nominatim_server`
: A character string specifying the base URL of the Nominatim server. Defaults to `"https://nominatim.openstreetmap.org/"`.

`custom_query`
: A named list of additional API parameters, for example `list(countrycodes = "US")`. See **Details**.

`points_only`
: A logical value indicating whether to return only point geometries. If `FALSE`, the API may return other geometry types. See **About geometry types**.

Details

A structured address search accepts an address already split into components. Each argument represents an address field. All components are optional, so provide only those relevant to the address you want to find.

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

Value

An `sf` object with the results that match the query.

About geometry types

The `points_only` argument controls whether the results contain only points. All Nominatim results have at least a point geometry.

When `points_only = FALSE`, the geometry type depends on the matching feature. Administrative areas and major buildings are returned as polygons, rivers and roads are returned as lines and amenities may still be returned as points.

This function is vectorized, allowing multiple addresses to be searched. With `points_only = FALSE`, multiple geometry types may be returned.

See Also

Address search functions: `geo_lite()`, `geo_lite_sf()`, `geo_lite_struct()`

Spatial helper and output functions: `bbox_to_poly()`, `geo_address_lookup_sf()`, `geo_amenity_sf()`, `geo_lite_sf()`, `reverse_geo_lite_sf()`

Examples

```
# Search with a structured address.

pl_mayor <- geo_lite_struct_sf(
  street = "Plaza Mayor",
  county = "Comunidad de Madrid",
  country = "Spain", limit = 50,
```

```
  full_results = TRUE, verbose = TRUE
)

# Retrieve an administrative boundary.
ccaa <- geo_lite_sf("Comunidad de Madrid, Spain", points_only = FALSE)

library(ggplot2)

if (any(!sf::st_is_empty(pl_mayor), !sf::st_is_empty(ccaa))) {
  ggplot(ccaa) +
    geom_sf() +
    geom_sf(data = pl_mayor, aes(shape = addresstype, color = addresstype))
}
```

`osm_amenities` *OpenStreetMap amenities*

Description

A dataset of amenity values available on OpenStreetMap.

Format

A tibble with 140 rows and three columns:

category Amenity category.

amenity Amenity value.

comment Brief description of the amenity type.

Note

The data were extracted on **July 11, 2026**. See `inst/COPYRIGHTS` for copyright and license details.

Source

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

See Also

Amenity lookup functions and data: `geo_amenity()`, `geo_amenity_sf()`

Examples

```
data("osm_amenities")

osm_amenities
```

`reverse_geo_lite` *Reverse geocode coordinates*

Description

Reverse geocodes latitude and longitude coordinates and returns matching results as a tibble. Latitude values must be in $[-90, 90]$ and longitudes in $[-180, 180]$. Use `reverse_geo_lite_sf()` to return an `sf` object instead.

Usage

```
reverse_geo_lite(
  lat,
  long,
  address = "address",
  full_results = FALSE,
  return_coords = TRUE,
  verbose = FALSE,
  nominatim_server = "https://nominatim.openstreetmap.org/",
  progressbar = TRUE,
  custom_query = list()
)
```

Arguments

`lat`	A numeric vector of latitude values in the range $[-90, 90]$.
`long`	A numeric vector of longitude values in the range $[-180, 180]$.
`address`	A character string specifying the name of the address column in the output. Defaults to `"address"`.
`full_results`	A logical value indicating whether to return all available fields from the Nominatim API. If `FALSE`, only query metadata, location data and requested address columns are returned.
`return_coords`	A logical value indicating whether to return the input coordinates with the results.
`verbose`	A logical value indicating whether to display detailed messages in the console.
`nominatim_server`	A character string specifying the base URL of the Nominatim server. Defaults to `"https://nominatim.openstreetmap.org/"`.
`progressbar`	A logical value indicating whether to display a progress bar when processing multiple queries.
`custom_query`	A named list of API-specific parameters, for example `list(zoom = 3)`. See **Details**.

Details

See https://nominatim.org/release-docs/latest/api/Reverse/ for additional parameters to pass to `custom_query`.

Value

A `tibble` with the results that match the query.

About zooming

Set `custom_query = list(zoom = 3)` to adjust the output. Selected zoom levels correspond to these address details:

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

See Also

`geo_lite()` for geocoding addresses.

Reverse geocoding functions: `reverse_geo_lite_sf()`

Examples

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

# Reverse geocode multiple coordinate pairs.
reverse_geo_lite(lat = c(40.75728, 55.95335), long = c(-73.98586, -3.188375))

# Set the zoom to the country level.
sev <- reverse_geo_lite(
  lat = c(40.75728, 55.95335), long = c(-73.98586, -3.188375),
  custom_query = list(zoom = 0, extratags = TRUE),
  verbose = TRUE, full_results = TRUE
)

dplyr::glimpse(sev)
```

reverse_geo_lite_sf *Reverse geocode coordinates and return* R*hrefhttps://CRAN.R-project.org/package=sf***sf** *objects*

Description

Reverse geocodes latitude and longitude coordinates and returns matching results as an `sf` object. Latitude values must be in $[-90, 90]$ and longitude values in $[-180, 180]$. Use `reverse_geo_lite()` to return a tibble instead.

Usage

```
reverse_geo_lite_sf(
  lat,
  long,
  address = "address",
  full_results = FALSE,
  return_coords = TRUE,
  verbose = FALSE,
  nominatim_server = "https://nominatim.openstreetmap.org/",
  progressbar = TRUE,
  custom_query = list(),
  points_only = TRUE
)
```

Arguments

`lat`	A numeric vector of latitude values in the range $[-90, 90]$.
`long`	A numeric vector of longitude values in the range $[-180, 180]$.
`address`	A character string specifying the name of the address column in the output. Defaults to `"address"`.
`full_results`	A logical value indicating whether to return all available fields from the Nominatim API. If `FALSE`, only query metadata, geometry and requested address columns are returned.
`return_coords`	A logical value indicating whether to return the input coordinates with the results.
`verbose`	A logical value indicating whether to display detailed messages in the console.
`nominatim_server`	A character string specifying the base URL of the Nominatim server. Defaults to `"https://nominatim.openstreetmap.org/"`.
`progressbar`	A logical value indicating whether to display a progress bar when processing multiple queries.
`custom_query`	A named list of API-specific parameters, for example `list(zoom = 3)`. See **Details**.
`points_only`	A logical value indicating whether to return only point geometries. If `FALSE`, the API may return other geometry types. See **About geometry types**.

Details

See https://nominatim.org/release-docs/latest/api/Reverse/ for additional parameters to pass to `custom_query`.

Value

An sf object with the results that match the query.

About zooming

Set `custom_query = list(zoom = 3)` to adjust the output. Selected zoom levels correspond to these address details:

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

About geometry types

The `points_only` argument controls whether the results contain only points. All Nominatim results have at least a point geometry.

When `points_only = FALSE`, the geometry type depends on the matching feature. Administrative areas and major buildings are returned as polygons, rivers and roads are returned as lines and amenities may still be returned as points.

This function is vectorized, allowing multiple addresses to be searched. With `points_only = FALSE`, multiple geometry types may be returned.

See Also

Reverse geocoding functions: `reverse_geo_lite()`

Spatial helper and output functions: `bbox_to_poly()`, `geo_address_lookup_sf()`, `geo_amenity_sf()`, `geo_lite_sf()`, `geo_lite_struct_sf()`

Examples

```
library(ggplot2)

# Define the Colosseum coordinates.
col_lon <- 12.49309
col_lat <- 41.89026
```

```
# Return the Colosseum as a polygon.
col_sf <- reverse_geo_lite_sf(
  lat = col_lat,
  long = col_lon,
  points_only = FALSE
)

dplyr::glimpse(col_sf)

if (!all(sf::st_is_empty(col_sf))) {
  ggplot(col_sf) +
    geom_sf()
}

# Return the city of Rome by using the same coordinates with zoom 10.

rome_sf <- reverse_geo_lite_sf(
  lat = col_lat,
  long = col_lon,
  custom_query = list(zoom = 10),
  points_only = FALSE
)

dplyr::glimpse(rome_sf)

if (!all(sf::st_is_empty(rome_sf))) {
  ggplot(rome_sf) +
    geom_sf()
}
```

Index

∗ **amenity**
  geo_amenity, 7
  geo_amenity_sf, 10
  osm_amenities, 20
∗ **datasets**
  osm_amenities, 20
∗ **geocoding**
  geo_lite, 12
  geo_lite_sf, 14
  geo_lite_struct, 16
  geo_lite_struct_sf, 18
∗ **lookup**
  geo_address_lookup, 4
  geo_address_lookup_sf, 5
∗ **reverse**
  reverse_geo_lite, 21
  reverse_geo_lite_sf, 23
∗ **spatial**
  bbox_to_poly, 2
  geo_address_lookup_sf, 5
  geo_amenity_sf, 10
  geo_lite_sf, 14
  geo_lite_struct_sf, 18
  reverse_geo_lite_sf, 23

amenities, *8*, *10*

bbox_to_poly, 2
bbox_to_poly(), *7*, *9*, *11*, *15*, *19*, *24*

geo_address_lookup, 4
geo_address_lookup(), *5*, *7*, *13*
geo_address_lookup_sf, 5
geo_address_lookup_sf(), *3–5*, *11*, *15*, *19*, *24*
geo_amenity, 7
geo_amenity(), *10, 11*, *13*, *16*, *18*, *20*
geo_amenity_sf, 10
geo_amenity_sf(), *3*, *7–9*, *15*, *19, 20*, *24*
geo_lite, 12
geo_lite(), *5*, *14–17*, *19*, *22*
geo_lite_sf, 14
geo_lite_sf(), *3*, *7*, *11–13*, *17–19*, *24*
geo_lite_struct, 16
geo_lite_struct(), *13*, *15*, *18, 19*
geo_lite_struct_sf, 18
geo_lite_struct_sf(), *3*, *7*, *11*, *13*, *15–17*, *24*

osm_amenities, *8–11*, 20

reverse_geo_lite, 21
reverse_geo_lite(), *13*, *23, 24*
reverse_geo_lite_sf, 23
reverse_geo_lite_sf(), *3*, *7*, *11*, *15*, *19*, *21*, *22*

sf, *4–6*, *8*, *10–12*, *14–16*, *18, 19*, *21*, *23, 24*
sf::st_as_sfc(), *3*
sf::st_sfc(), *3*
sfc, *2, 3*, *8*, *10*
st_crs, *3*

tibble, *4, 5*, *8–10*, *12–14*, *16–18*, *20–23*