Package: opencage (via r-universe)

October 31, 2026

Type Package

Title Geocode with the OpenCage API

Version 0.3.0

Description Geocode with the OpenCage API, either from place name to longitude and latitude (forward geocoding) or from longitude and latitude to the name and address of a location (reverse geocoding), see <https://opencagedata.com>.

License GPL (>= 2)

URL https://docs.ropensci.org/opencage/, https://github.com/ropensci/opencage

BugReports https://github.com/ropensci/opencage/issues

Depends R (>= 3.4.0)

Imports crul (>= 0.5.2), dplyr (>= 0.7.4), jsonlite (>= 1.5), lifecycle, memoise (>= 1.1.0), progress (>= 1.1.2), purrr (>= 0.2.4), ratelimitr (>= 0.4.0), rlang, tibble (>= 1.4.2), tidyr (>= 1.0.0), withr (>= 2.0.0)

Suggests knitr (>= 1.19), mockery, rmarkdown (>= 1.8), sf (>= 0.9), testthat (>= 3.1.0)

VignetteBuilder knitr

RdMacros lifecycle

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData TRUE

NeedsCompilation no

Author Daniel Possenriede [aut, cre] (ORCID: <https://orcid.org/0000-0002-6738-9845>), Jesse Sadler [aut] (ORCID: <https://orcid.org/0000-0001-6081-9681>), Maëlle Salmon

[aut] (ORCID: <https://orcid.org/0000-0002-2815-0399>), Noam Ross [ctb], Jake Russ [ctb], Julia Silge [rev] (Julia Silge reviewed \{opencage\} (v. 0.1.0) for rOpenSci, see <https://github.com/ropensci/onboarding/issues/36>.)

Maintainer Daniel Possenriede <possenriede+r@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-10-31 17:50:04 UTC

RemoteUrl https://github.com/cran/opencage

RemoteRef HEAD

RemoteSha e890e90f38458238be94e8a2c67152bd490c3bee

Contents

countrycodes 2
oc_bbox 3
oc_clear_cache 4
oc_config 5
oc_forward 6
oc_forward_df 9
oc_points 13
oc_reverse 14
oc_reverse_df 16
opencage-deprecated 19
opencage_forward 19
opencage_reverse 21

Index **23**

`countrycodes` *Country codes*

Description

Country codes

Format

All possible ISO 3166-1 Alpha 2 standard country codes.

Examples

```
data("countrycodes")
```

oc_bbox *List of bounding boxes for OpenCage queries*

Description

Create a list of bounding boxes for OpenCage queries.

Usage

```
oc_bbox(...)

## S3 method for class 'numeric'
oc_bbox(xmin, ymin, xmax, ymax, ...)

## S3 method for class 'data.frame'
oc_bbox(data, xmin, ymin, xmax, ymax, ...)

## S3 method for class 'bbox'
oc_bbox(bbox, ...)
```

Arguments

`...`	Ignored.
`xmin`	Minimum longitude (also known as `min_lon`, `southwest_lng`, `west`, or `left`).
`ymin`	Minimum latitude (also known as `min_lat`, `southwest_lat`, `south`, or `bottom`).
`xmax`	Maximum longitude (also known as `max_lon`, `northeast_lng`, `east`, or `right`).
`ymax`	Maximum latitude (also known as `max_lat`, `northeast_lat`, `north`, or `top`).
`data`	A `data.frame` containing at least 4 columns with `xmin`, `ymin`, `xmax`, and `ymax` values, respectively.
`bbox`	A `bbox` object, see `sf::st_bbox`.

Value

A list of bounding boxes, each of class `bbox`.

Examples

```
oc_bbox(-5.63160, 51.280430, 0.278970, 51.683979)

xdf <-
  data.frame(
    place = c("Hamburg", "Hamburg"),
    northeast_lat = c(54.0276817, 42.7397729),
    northeast_lng = c(10.3252805, -78.812825),
    southwest_lat = c(53.3951118, 42.7091669),
    southwest_lng = c(8.1053284, -78.860521)
```

```
  )
oc_bbox(
  xdf,
  southwest_lng,
  southwest_lat,
  northeast_lng,
  northeast_lat
)

# create bbox list column with dplyr
library(dplyr)
xdf %>%
  mutate(
    bbox =
      oc_bbox(
        southwest_lng,
        southwest_lat,
        northeast_lng,
        northeast_lat
      )
  )

# create bbox list from a simple features bbox
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  bbox <- st_bbox(c(xmin = 16.1, xmax = 16.6, ymax = 48.6, ymin = 47.9),
    crs = 4326
  )
  oc_bbox(bbox)
}
```

`oc_clear_cache` *Clear the opencage cache*

Description

Forget past results and reset the **opencage** cache.

Usage

```
oc_clear_cache()
```

Examples

```
system.time(oc_reverse(latitude = 10, longitude = 10))
system.time(oc_reverse(latitude = 10, longitude = 10))
oc_clear_cache()
system.time(oc_reverse(latitude = 10, longitude = 10))
```

`oc_config` *Configure settings*

Description

Configure session settings for **opencage**.

Usage

```
oc_config(
  key = Sys.getenv("OPENCAGE_KEY"),
  rate_sec = getOption("oc_rate_sec", default = 1L),
  no_record = getOption("oc_no_record", default = TRUE),
  show_key = getOption("oc_show_key", default = FALSE),
  ...
)
```

Arguments

`key`	Your OpenCage API key as a character vector of length one. Do not pass the key directly as a parameter, though. See details.
`rate_sec`	Numeric vector of length one. Sets the maximum number of requests sent to the OpenCage API per second. Defaults to the value set in the `oc_rate_sec` option, or, in case that does not exist, to 1L.
`no_record`	Logical vector of length one. When `TRUE`, OpenCage will not create log entries of the queries and will not cache the geocoding requests. Defaults to the value set in the `oc_no_record` option, or, in case that does not exist, to `TRUE`.
`show_key`	Logical vector of length one. This is only relevant when debugging `oc_forward()` or `oc_reverse()` calls with the `return = "url_only"` argument. When `TRUE`, the result will show your OpenCage API key in the URL as stored in the `OPENCAGE_KEY` environment variable. When not `TRUE`, the API key will be replaced with the string `OPENCAGE_KEY`. `show_key` defaults to the value set in the `oc_show_key` option, or, in case that does not exist, to `FALSE`.
`...`	Ignored.

Set your OpenCage API key

opencage will conveniently retrieve your API key if it is saved in the environment variable `"OPENCAGE_KEY"`. `oc_config()` will help to set that environment variable. Do not pass the key directly as a parameter to the function, though, as you risk exposing it via your script or your history. There are three safer ways to set your API key instead:

1. Save your API key as an environment variable in `.Renviron` as described in What They Forgot to Teach You About R or Efficient R Programming. From there it will be fetched by all functions that call the OpenCage API. You do not even have to call `oc_config()` to set your key; you can start geocoding right away. If you have the **usethis** package installed, you can

edit your .Renviron with usethis::edit_r_environ(). We strongly recommend storing your API key in the user-level .Renviron, as opposed to the project-level. This makes it less likely you will share sensitive information by mistake.

2. If you use a package like **keyring** to store your credentials, you can safely pass your key in a script with a function call like this oc_config(key = keyring::key_get("opencage")).
3. If you call oc_config() in an base::interactive() session and the OPENCAGE_KEY environment variable is not set, it will prompt you to enter the key in the console.

Set your OpenCage API rate limit

The rate limit allowed by the API depends on the OpenCage plan you purchased and ranges from 1 request/sec for the "Free Trial" plan to 15 requests/sec for the "Medium" or "Large" plans, see https://opencagedata.com/pricing for details and up-to-date information. You can set the rate limit persistently across sessions by setting an oc_rate_sec option in your .Rprofile. If you have the **usethis** package installed, you can edit your .Rprofile with usethis::edit_r_profile().

Prevent query logging and caching

By default, OpenCage will store your queries in its server logs and will cache the forward geocoding requests on their side. They do this in order to speed up response times and to be able to debug errors and improve their service. Logs are automatically deleted after six months according to OpenCage's page on data protection and GDPR.

If you set no_record to TRUE, the query contents are not logged nor cached. OpenCage still records that you made a request, but not the specific query you made. oc_config(no_record = TRUE) sets the oc_no_record option for the active R session, so it will be used for all subsequent OpenCage queries. You can set the oc_no_record option persistently across sessions in your .Rprofile.

For increased privacy **opencage** sets no_record to TRUE, by default. Please note, however, that **opencage** always caches the data it receives from the OpenCage API locally, but only for as long as your R session is alive.

For more information on OpenCage's policies on privacy and data protection see their FAQs, their GDPR page, and, for the no_record parameter, see the relevant blog post.

---

oc_forward *Forward geocoding*

Description

Forward geocoding from a character vector of location names to latitude and longitude tuples.

Usage

```
oc_forward(
  placename,
  return = c("df_list", "json_list", "geojson_list", "url_only"),
  bounds = NULL,
  proximity = NULL,
```

```
  countrycode = NULL,
  language = NULL,
  limit = 10L,
  no_annotations = TRUE,
  roadinfo = FALSE,
  no_dedupe = FALSE,
  abbrv = FALSE,
  address_only = FALSE,
  add_request = FALSE,
  ...
)
```

Arguments

placename	A character vector with the location names or addresses to be geocoded. If the locations are addresses, see OpenCage's instructions on how to format addresses for best forward geocoding results.
return	A character vector of length one indicating the return value of the function, either a list of tibbles (`df_list`, the default), a JSON list (`json_list`), a GeoJSON list (`geojson_list`), or the URL with which the API would be called (`url_only`).
bounds	A list of bounding boxes of length one or `length(placename)`. Bounding boxes are named numeric vectors, each with four coordinates forming the south-west and north-east corners of the bounding box: `list(c(xmin, ymin, xmax, ymax))`. `bounds` restricts the possible results to the supplied region. It can be specified with the `oc_bbox()` helper. For example: `bounds = oc_bbox(-0.563160, 51.280430, 0.278970, 51.683979)`. Default is `NULL`.
proximity	A list of points of length one or `length(placename)`. A point is a named numeric vector of a latitude, longitude coordinate pair in decimal format. `proximity` provides OpenCage with a hint to bias results in favour of those closer to the specified location. It can be specified with the `oc_points()` helper. For example: `proximity = oc_points(51.9526, 7.6324)`. Default is `NULL`.
countrycode	A two letter code as defined by the ISO 3166-1 Alpha 2 standard that restricts the results to the given country or countries. E.g. "AR" for Argentina, "FR" for France, "NZ" for the New Zealand. Multiple countrycodes per `placename` must be wrapped in a list. Default is `NULL`.
language	An IETF BCP 47 language tag (such as "es" for Spanish or "pt-BR" for Brazilian Portuguese). OpenCage will attempt to return results in that language. Alternatively you can specify the "native" tag, in which case OpenCage will attempt to return the response in the "official" language(s). In case the `language` parameter is set to `NULL` (which is the default), the tag is not recognized, or OpenCage does not have a record in that language, the results will be returned in English.
limit	Numeric vector of integer values to determine the maximum number of results returned for each `placename`. Integer values between 1 and 100 are allowed. Default is 10.
no_annotations	Logical vector indicating whether additional information about the result location should be returned. `TRUE` by default, which means that the results will not contain annotations.

roadinfo	Logical vector indicating whether the geocoder should attempt to match the nearest road (rather than an address) and provide additional road and driving information. Default is FALSE.
no_dedupe	Logical vector (default FALSE), when TRUE the results will not be deduplicated.
abbrv	Logical vector (default FALSE), when TRUE addresses in the formatted field of the results are abbreviated (e.g. "Main St." instead of "Main Street").
address_only	Logical vector (default FALSE), when TRUE only the address details are returned in the formatted field of the results, not the name of a point-of-interest should there be one at this address.
add_request	Logical vector (default FALSE) indicating whether the request is returned again with the results. If the return value is a df_list, the query text is added as a column to the results. json_list results will contain all request parameters, including the API key used! This is currently ignored by OpenCage if return value is geojson_list.
...	Ignored.

Value

Depending on the return argument, oc_forward returns a list with either

- the results as tibbles ("df_list", the default),
- the results as JSON specified as a list ("json_list"),
- the results as GeoJSON specified as a list ("geojson_list"), or
- the URL of the OpenCage API call for debugging purposes ("url_only").

When the results are returned as (a list of) tibbles, the column names coming from the OpenCage API are prefixed with "oc_".

See Also

oc_forward_df() for inputs as a data frame, or oc_reverse() and oc_reverse_df() for reverse geocoding. For more information about the API and the various parameters, see the OpenCage API documentation.

Examples

```
# Geocode a single location, an address in this case
oc_forward(placename = "Triererstr 15, 99432, Weimar, Deutschland")

# Geocode multiple locations
locations <- c("Nantes", "Hamburg", "Los Angeles")
oc_forward(placename = locations)

# Use bounding box to help return accurate results
# for each placename
bounds <- oc_bbox(
  xmin = c(-2, 9, -119),
```

```
  ymin = c(47, 53, 34),
  xmax = c(0, 10, -117),
  ymax = c(48, 54, 35)
)
oc_forward(placename = locations, bounds = bounds)

# Another way to help specify the desired results
# is with country codes.
oc_forward(
  placename = locations,
  countrycode = c("ca", "us", "co")
)

# With multiple countrycodes per placename
oc_forward(
  placename = locations,
  countrycode = list(c("fr", "ca"), c("de", "us"), c("us", "co"))
)

# Return results in a preferred language if possible
oc_forward(
  placename = c("Brugge", "Mechelen", "Antwerp"),
  language = "fr"
)

# Limit the number of results per placename and return json_list
oc_forward(
  placename = locations,
  bounds = bounds,
  limit = 1,
  return = "json_list"
)
```

`oc_forward_df` *Forward geocoding with data frames*

Description

Forward geocoding from a column or vector of location names to latitude and longitude tuples.

Usage

```
oc_forward_df(...)

## S3 method for class 'data.frame'
oc_forward_df(
  data,
  placename,
  bind_cols = TRUE,
```

```
  output = c("short", "all"),
  bounds = NULL,
  proximity = NULL,
  countrycode = NULL,
  language = NULL,
  limit = 1L,
  no_annotations = TRUE,
  roadinfo = FALSE,
  no_dedupe = FALSE,
  abbrv = FALSE,
  address_only = FALSE,
  ...
)

## S3 method for class 'character'
oc_forward_df(
  placename,
  output = c("short", "all"),
  bounds = NULL,
  proximity = NULL,
  countrycode = NULL,
  language = NULL,
  limit = 1L,
  no_annotations = TRUE,
  roadinfo = FALSE,
  no_dedupe = FALSE,
  abbrv = FALSE,
  address_only = FALSE,
  ...
)
```

Arguments

`...`	Ignored.
`data`	A data frame.
`placename`	An unquoted variable name of a character column or vector with the location names or addresses to be geocoded. If the locations are addresses, see OpenCage's instructions on how to format addresses for best forward geocoding results.
`bind_cols`	When `bind_col = TRUE`, the default, the results are column bound to `data`. When `FALSE`, the results are returned as a new tibble.
`output`	A character vector of length one indicating whether only latitude, longitude, and formatted address variables (`"short"`, the default), or all variables (`"all"`) variables should be returned.
`bounds`	A list of length one, or an unquoted variable name of a list column of bounding boxes. Bounding boxes are named numeric vectors, each with 4 coordinates forming the south-west and north-east corners of the bounding box:

list(c(xmin, ymin, xmax, ymax)). bounds restricts the possible results to the supplied region. It can be specified with the oc_bbox() helper. For example: bounds = oc_bbox(-0.563160, 51.280430, 0.278970, 51.683979). Default is NULL.

proximity — A list of length one, or an unquoted variable name of a list column of points. Points are named numeric vectors with latitude, longitude coordinate pairs in decimal format. proximity provides OpenCage with a hint to bias results in favour of those closer to the specified location. It can be specified with the oc_points() helper. For example: proximity = oc_points(41.40139, 2.12870). Default is NULL.

countrycode — Character vector, or an unquoted variable name of such a vector, of two-letter codes as defined by the ISO 3166-1 Alpha 2 standard that restricts the results to the given country or countries. E.g. "AR" for Argentina, "FR" for France, "NZ" for the New Zealand. Multiple countrycodes per placename must be wrapped in a list. Default is NULL.

language — Character vector, or an unquoted variable name of such a vector, of IETF BCP 47 language tags (such as "es" for Spanish or "pt-BR" for Brazilian Portuguese). OpenCage will attempt to return results in that language. Alternatively you can specify the "native" tag, in which case OpenCage will attempt to return the response in the "official" language(s). In case the language parameter is set to NULL (which is the default), the tag is not recognized, or OpenCage does not have a record in that language, the results will be returned in English.

limit — Numeric vector of integer values, or an unquoted variable name of such a vector, to determine the maximum number of results returned for each placename. Integer values between 1 and 100 are allowed. Default is 1.

no_annotations — Logical vector, or an unquoted variable name of such a vector, indicating whether additional information about the result location should be returned. TRUE by default, which means that the results will not contain annotations.

roadinfo — Logical vector, or an unquoted variable name of such a vector, indicating whether the geocoder should attempt to match the nearest road (rather than an address) and provide additional road and driving information. Default is FALSE.

no_dedupe — Logical vector, or an unquoted variable name of such a vector. Default is FALSE. When TRUE the results will not be deduplicated.

abbrv — Logical vector, or an unquoted variable name of such a vector. Default is FALSE. When TRUE addresses in the oc_formatted variable of the results are abbreviated (e.g. "Main St." instead of "Main Street").

address_only — Logical vector, or an unquoted variable name of such a vector. Default is FALSE. When TRUE only the address details are returned in the oc_formatted variable of the results, not the name of a point-of-interest should there be one at this address.

Value

A tibble. Column names coming from the OpenCage API are prefixed with "oc_".

See Also

`oc_forward()` for inputs as vectors, or `oc_reverse()` and `oc_reverse_df()` for reverse geocoding. For more information about the API and the various parameters, see the OpenCage API documentation.

Examples

```
library(tibble)
df <- tibble(
  id = 1:3,
  locations = c("Nantes", "Hamburg", "Los Angeles")
)

# Return lat, lng, and formatted address
oc_forward_df(df, placename = locations)

# Return more detailed information about the locations
oc_forward_df(df, placename = locations, output = "all")

# Do not column bind results to input data frame
oc_forward_df(df, placename = locations, bind_cols = FALSE)

# Add more results by changing the limit from the default of 1.
oc_forward_df(df, placename = locations, limit = 5)

# Restrict results to a given bounding box
oc_forward_df(df,
  placename = locations,
  bounds = oc_bbox(-5, 45, 15, 55)
)

# oc_forward_df accepts unquoted column names for all
# arguments except bind_cols and output.
# This makes it possible to build up more detailed queries
# through the data frame passed to the data argument.

df2 <- add_column(df,
  bounds = oc_bbox(
    xmin = c(-2, 9, -119),
    ymin = c(47, 53, 34),
    xmax = c(0, 10, -117),
    ymax = c(48, 54, 35)
  ),
  limit = 1:3,
  countrycode = c("ca", "us", "co"),
  language = c("fr", "de", "en")
)

# Use the bounds column to help return accurate results and
# language column to specify preferred language of results
oc_forward_df(df2,
```

```
  placename = locations,
  bounds = bounds,
  language = language
)

# Different limit of results for each placename
oc_forward_df(df2,
  placename = locations,
  limit = limit
)

# Specify the desired results by the countrycode column
oc_forward_df(df2,
  placename = locations,
  countrycode = countrycode
)
```

`oc_points` *List of points for OpenCage queries*

Description

Create a list of points (latitude/longitude coordinate pairs) for OpenCage queries.

Usage

```
oc_points(...)

## S3 method for class 'numeric'
oc_points(latitude, longitude, ...)

## S3 method for class 'data.frame'
oc_points(data, latitude, longitude, ...)
```

Arguments

`...`	Ignored.
`latitude, longitude`	Numeric vectors of latitude and longitude values.
`data`	A `data.frame` containing at least 2 columns with `latitude` and `longitude` values.

Value

A list of points. Each point is a named vector of length 2 containing a latitude/longitude coordinate pair.

Examples

```
oc_points(-21.01404, 55.26077)

xdf <-
  data.frame(
    place = c("Hamburg", "Los Angeles"),
    lat = c(53.5503, 34.0536),
    lon = c(10.0006, -118.2427)
  )
oc_points(
  data = xdf,
  latitude = lat,
  longitude = lon
)

# create a list column with points with dplyr
library(dplyr)
xdf %>%
  mutate(
    points =
      oc_points(
        lat,
        lon
      )
  )
```

`oc_reverse` *Reverse geocoding*

Description

Reverse geocoding from numeric vectors of latitude and longitude pairs to the names and addresses of a location.

Usage

```
oc_reverse(
  latitude,
  longitude,
  return = c("df_list", "json_list", "geojson_list", "url_only"),
  language = NULL,
  no_annotations = TRUE,
  roadinfo = FALSE,
  no_dedupe = FALSE,
  abbrv = FALSE,
  address_only = FALSE,
  add_request = FALSE,
  ...
)
```

Arguments

`latitude, longitude`
: Numeric vectors of latitude and longitude values.

`return`
: A character vector of length one indicating the return value of the function, either a list of tibbles (`df_list`, the default), a JSON list (`json_list`), a GeoJSON list (`geojson_list`), or the URL with which the API would be called (`url_only`).

`language`
: An IETF BCP 47 language tag (such as "es" for Spanish or "pt-BR" for Brazilian Portuguese). OpenCage will attempt to return results in that language. Alternatively you can specify the "native" tag, in which case OpenCage will attempt to return the response in the "official" language(s). In case the `language` parameter is set to `NULL` (which is the default), the tag is not recognized, or OpenCage does not have a record in that language, the results will be returned in English.

`no_annotations`
: Logical vector indicating whether additional information about the result location should be returned. `TRUE` by default, which means that the results will not contain annotations.

`roadinfo`
: Logical vector indicating whether the geocoder should attempt to match the nearest road (rather than an address) and provide additional road and driving information. Default is `FALSE`.

`no_dedupe`
: Logical vector (default `FALSE`), when `TRUE` the results will not be deduplicated.

`abbrv`
: Logical vector (default `FALSE`), when `TRUE` addresses in the `formatted` field of the results are abbreviated (e.g. "Main St." instead of "Main Street").

`address_only`
: Logical vector (default `FALSE`), when `TRUE` only the address details are returned in the `formatted` field of the results, not the name of a point-of-interest should there be one at this address.

`add_request`
: Logical vector (default `FALSE`) indicating whether the request is returned again with the results. If the `return` value is a `df_list`, the query text is added as a column to the results. `json_list` results will contain all request parameters, including the API key used! This is currently ignored by OpenCage if return value is `geojson_list`.

`...`
: Ignored.

Value

Depending on the `return` argument, `oc_reverse` returns a list with either

- the results as tibbles (`"df_list"`, the default),
- the results as JSON specified as a list (`"json_list"`),
- the results as GeoJSON specified as a list (`"geojson_list"`), or
- the URL of the OpenCage API call for debugging purposes (`"url_only"`).

When the results are returned as (a list of) tibbles, the column names coming from the OpenCage API are prefixed with `"oc_"`.

See Also

`oc_reverse_df()` for inputs as a data frame, or `oc_forward()` and `oc_forward()` for forward geocoding. For more information about the API and the various parameters, see the OpenCage API documentation.

Examples

```
# Reverse geocode a single location
oc_reverse(latitude = -36.85007, longitude = 174.7706)

# Reverse geocode multiple locations
lat <- c(47.21864, 53.55034, 34.05369)
lng <- c(-1.554136, 10.000654, -118.242767)

oc_reverse(latitude = lat, longitude = lng)

# Return results in a preferred language if possible
oc_reverse(
  latitude = lat, longitude = lng,
  language = "fr"
)

# Return results as a json list
oc_reverse(
  latitude = lat, longitude = lng,
  return = "json_list"
)
```

---

oc_reverse_df *Reverse geocoding with data frames*

Description

Reverse geocoding from latitude and longitude pairs to the names and addresses of a location.

Usage

```
oc_reverse_df(...)

## S3 method for class 'data.frame'
oc_reverse_df(
  data,
  latitude,
  longitude,
  bind_cols = TRUE,
  output = c("short", "all"),
```

```
  language = NULL,
  roadinfo = FALSE,
  no_annotations = TRUE,
  no_dedupe = FALSE,
  abbrv = FALSE,
  address_only = FALSE,
  ...
)

## S3 method for class 'numeric'
oc_reverse_df(
  latitude,
  longitude,
  output = c("short", "all"),
  language = NULL,
  no_annotations = TRUE,
  no_dedupe = FALSE,
  abbrv = FALSE,
  address_only = FALSE,
  ...
)
```

Arguments

`...`	Ignored.
`data`	A data frame.
`latitude, longitude`	Unquoted variable names of numeric columns or vectors of latitude and longitude values.
`bind_cols`	When `bind_col = TRUE`, the default, the results are column bound to `data`. When `FALSE`, the results are returned as a new tibble.
`output`	A character vector of length one indicating whether only the formatted address (`"short"`, the default) or all variables (`"all"`) variables should be returned.
`language`	Character vector, or an unquoted variable name of such a vector, of IETF BCP 47 language tags (such as "es" for Spanish or "pt-BR" for Brazilian Portuguese). OpenCage will attempt to return results in that language. Alternatively you can specify the "native" tag, in which case OpenCage will attempt to return the response in the "official" language(s). In case the `language` parameter is set to `NULL` (which is the default), the tag is not recognized, or OpenCage does not have a record in that language, the results will be returned in English.
`roadinfo`	Logical vector, or an unquoted variable name of such a vector, indicating whether the geocoder should attempt to match the nearest road (rather than an address) and provide additional road and driving information. Default is `FALSE`.
`no_annotations`	Logical vector, or an unquoted variable name of such a vector, indicating whether additional information about the result location should be returned. `TRUE` by default, which means that the results will not contain annotations.

no_dedupe	Logical vector, or an unquoted variable name of such a vector. Default is FALSE. When TRUE the results will not be deduplicated.
abbrv	Logical vector, or an unquoted variable name of such a vector. Default is FALSE. When TRUE addresses in the oc_formatted variable of the results are abbreviated (e.g. "Main St." instead of "Main Street").
address_only	Logical vector, or an unquoted variable name of such a vector. Default is FALSE. When TRUE only the address details are returned in the oc_formatted variable of the results, not the name of a point-of-interest should there be one at this address.

Value

A tibble. Column names coming from the OpenCage API are prefixed with "oc_".

See Also

oc_reverse() for inputs as vectors, or oc_forward() and oc_forward() for forward geocoding. For more information about the API and the various parameters, see the OpenCage API documentation.

Examples

```
library(tibble)
df <- tibble(
  id = 1:4,
  lat = c(-36.85007, 47.21864, 53.55034, 34.05369),
  lng = c(174.7706, -1.554136, 10.000654, -118.242767)
)

# Return formatted address of lat/lng values
oc_reverse_df(df, latitude = lat, longitude = lng)

# Return more detailed information about the locations
oc_reverse_df(df,
  latitude = lat, longitude = lng,
  output = "all"
)

# Return results in a preferred language if possible
oc_reverse_df(df,
  latitude = lat, longitude = lng,
  language = "fr"
)

# oc_reverse_df accepts unquoted column names for all
# arguments except bind_cols and output.
# This makes it possible to build up more detailed queries
# through the data frame passed to the data argument.
df2 <- add_column(df,
  language = c("en", "fr", "de", "en")
```

```
)

# Use language column to specify preferred language of results
oc_reverse_df(df2,
  latitude = lat, longitude = lng,
  language = language
)
```

---

opencage-deprecated *Deprecated functions in opencage*

Description

These functions still work but will be removed (defunct) in the next version.

Details

- opencage_forward()
- opencage_reverse()
- opencage_key()

---

opencage_forward *Forward geocoding*

Description

[Deprecated]

Deprecated: use `oc_forward` or `oc_forward_df` for forward geocoding.

Usage

```
opencage_forward(
  placename,
  key = opencage_key(),
  bounds = NULL,
  countrycode = NULL,
  language = NULL,
  limit = 10L,
  min_confidence = NULL,
  no_annotations = FALSE,
  no_dedupe = FALSE,
  no_record = FALSE,
  abbrv = FALSE,
  add_request = TRUE
)
```

Arguments

placename	A character vector with the location names or addresses to be geocoded. If the locations are addresses, see OpenCage's instructions on how to format addresses for best forward geocoding results.
key	Your OpenCage API key as a character vector of length one. By default, opencage_key() will attempt to retrieve the key from the environment variable OPENCAGE_KEY.
bounds	A list of bounding boxes of length one or length(placename). Bounding boxes are named numeric vectors, each with four coordinates forming the south-west and north-east corners of the bounding box: list(c(xmin, ymin, xmax, ymax)). bounds restricts the possible results to the supplied region. It can be specified with the oc_bbox() helper. For example: bounds = oc_bbox(-0.563160, 51.280430, 0.278970, 51.683979). Default is NULL.
countrycode	A two letter code as defined by the ISO 3166-1 Alpha 2 standard that restricts the results to the given country or countries. E.g. "AR" for Argentina, "FR" for France, "NZ" for the New Zealand. Multiple countrycodes per placename must be wrapped in a list. Default is NULL.
language	An IETF BCP 47 language tag (such as "es" for Spanish or "pt-BR" for Brazilian Portuguese). OpenCage will attempt to return results in that language. Alternatively you can specify the "native" tag, in which case OpenCage will attempt to return the response in the "official" language(s). In case the language parameter is set to NULL (which is the default), the tag is not recognized, or OpenCage does not have a record in that language, the results will be returned in English.
limit	Numeric vector of integer values to determine the maximum number of results returned for each placename. Integer values between 1 and 100 are allowed. Default is 10.
min_confidence	**[Defunct]** min_confidence is no longer supported by the OpenCage API.
no_annotations	Logical vector indicating whether additional information about the result location should be returned. TRUE by default, which means that the results will not contain annotations.
no_dedupe	Logical vector (default FALSE), when TRUE the results will not be deduplicated.
no_record	Logical vector of length one (default FALSE), when TRUE no log entry of the query is created, and the geocoding request is not cached by OpenCage.
abbrv	Logical vector (default FALSE), when TRUE addresses in the formatted field of the results are abbreviated (e.g. "Main St." instead of "Main Street").
add_request	Logical vector (default FALSE) indicating whether the request is returned again with the results. If the return value is a df_list, the query text is added as a column to the results. json_list results will contain all request parameters, including the API key used! This is currently ignored by OpenCage if return value is geojson_list.

Value

A list with

- results as a tibble with one line per result,

- the number of results as an integer,
- the timestamp as a POSIXct object,
- rate_info tibble/data.frame with the maximal number of API calls per day for the used key, the number of remaining calls for the day and the time at which the number of remaining calls will be reset.

Examples

```
opencage_forward(placename = "Sarzeau")
opencage_forward(placename = "Islington, London")
opencage_forward(placename = "Triererstr 15,
                              Weimar 99423,
                              Deutschland")
```

`opencage_reverse` *Reverse geocoding*

Description

[Deprecated]

Deprecated: use `oc_reverse` or `oc_reverse_df` for reverse geocoding.

Usage

```
opencage_reverse(
  latitude,
  longitude,
  key = opencage_key(),
  bounds = NULL,
  countrycode = NULL,
  language = NULL,
  limit = 10,
  min_confidence = NULL,
  no_annotations = FALSE,
  no_dedupe = FALSE,
  no_record = FALSE,
  abbrv = FALSE,
  add_request = TRUE
)
```

Arguments

`latitude, longitude`
: Numeric vectors of latitude and longitude values.

`key`
: Your OpenCage API key as a character vector of length one. By default, `opencage_key()` will attempt to retrieve the key from the environment variable `OPENCAGE_KEY`.

bounds	Bounding box, ignored for reverse geocoding.
countrycode	Country code, ignored for reverse geocoding.
language	An IETF BCP 47 language tag (such as "es" for Spanish or "pt-BR" for Brazilian Portuguese). OpenCage will attempt to return results in that language. Alternatively you can specify the "native" tag, in which case OpenCage will attempt to return the response in the "official" language(s). In case the language parameter is set to NULL (which is the default), the tag is not recognized, or OpenCage does not have a record in that language, the results will be returned in English.
limit	How many results should be returned (1-100), ignored for reverse geocoding.
min_confidence	**[Defunct]** min_confidence is no longer supported by the OpenCage API.
no_annotations	Logical vector indicating whether additional information about the result location should be returned. TRUE by default, which means that the results will not contain annotations.
no_dedupe	Logical vector (default FALSE), when TRUE the results will not be deduplicated.
no_record	Logical vector of length one (default FALSE), when TRUE no log entry of the query is created, and the geocoding request is not cached by OpenCage.
abbrv	Logical vector (default FALSE), when TRUE addresses in the formatted field of the results are abbreviated (e.g. "Main St." instead of "Main Street").
add_request	Logical vector (default FALSE) indicating whether the request is returned again with the results. If the return value is a df_list, the query text is added as a column to the results. json_list results will contain all request parameters, including the API key used! This is currently ignored by OpenCage if return value is geojson_list.

Value

A list with

- results as a tibble with one line per result,
- the number of results as an integer,
- the timestamp as a POSIXct object,
- rate_info tibble/data.frame with the maximal number of API calls per day for the used key, the number of remaining calls for the day and the time at which the number of remaining calls will be reset.

Examples

```
opencage_reverse(
  latitude = 0, longitude = 0,
  limit = 2
)
```

Index

∗ **data**
countrycodes, 2
.Renviron, *5, 6*
.Rprofile, *6*

base::interactive(), *6*

countrycodes, 2

oc_bbox, 3
oc_bbox(), *7*, *11*, *20*
oc_clear_cache, 4
oc_config, 5
oc_config(), *5*
oc_forward, 6
oc_forward(), *12*, *16*, *18*
oc_forward_df, 9
oc_forward_df(), *8*
oc_points, 13
oc_points(), *7*, *11*
oc_reverse, 14
oc_reverse(), *8*, *12*, *18*
oc_reverse_df, 16
oc_reverse_df(), *8*, *12*, *16*
opencage-deprecated, 19
opencage_forward, 19
opencage_forward(), *19*
opencage_key(), *19–21*
opencage_reverse, 21
opencage_reverse(), *19*
option, *6*