

# Package ‘opencage’

May 12, 2016

**Type** Package

**Title** Interface to the OpenCage API

**Version** 0.1.0

**Description** Tool for accessing the OpenCage API, which provides forward geocoding (from placename to longitude and latitude) and reverse geocoding (from longitude and latitude to placename).

**License** GPL (>= 2)

**LazyData** TRUE

**URL** <http://github.com/ropenscilabs/opencage>

**BugReports** <http://github.com/ropenscilabs/opencage/issues>

**Imports** httr, jsonlite, dplyr, memoise

**RoxygenNote** 5.0.1

**Suggests** testthat, lintr, knitr, rmarkdown

**VignetteBuilder** knitr

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Maëlle Salmon [aut, cre],  
Noam Ross [ctb],  
Julia Silge [ctb] (Julia Silge reviewed the package for rOpenSci, see  
<https://github.com/ropensci/onboarding/issues/36>.)

**Maintainer** Maëlle Salmon <maelle.salmon@yahoo.se>

**Repository** CRAN

**Date/Publication** 2016-05-12 15:15:13

## R topics documented:

code_message . . . . .	2
countrycodes . . . . .	2
languagecodes . . . . .	2
opencage_forward . . . . .	3
opencage_reverse . . . . .	4

**Index****6**


---

code_message	<i>API messages</i>
--------------	---------------------

---

**Description**

API messages

**Format**

Codes and corresponding messages from the API.

**Examples**

```
data("code_message")
```

---

countrycodes	<i>Country codes</i>
--------------	----------------------

---

**Description**

Country codes

**Format**

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

**Examples**

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

---

languagecodes	<i>Language codes</i>
---------------	-----------------------

---

**Description**

Language codes

**Format**

All possible ISO 639-2 language codes.

**Examples**

```
data("languagecodes")
```

---

opencage_forward	<i>Forward geocoding</i>
------------------	--------------------------

---

### Description

Forward geocoding, from placename to latitude and longitude tuple(s).

### Usage

```
opencage_forward(placename, key = opencage_key(), bounds = NULL,
  countrycode = NULL, language = NULL, limit = 10,
  min_confidence = NULL, no_annotations = NULL, no_dedupe = NULL)
```

### Arguments

placename	Placename.
key	Your OpenCage key.
bounds	Provides the geocoder with a hint to the region that the query resides in. This value will restrict the possible results to the supplied region. The bounds parameter should be specified as 4 coordinate points forming the south-west and north-east corners of a bounding box. For example, bounds = c(-0.563160, 51.280430, 0.278970, 51.683160) (min long, min lat, max long, max lat).
countrycode	Restricts the results to the given country. The country code is a two letter code as defined by the ISO 3166-1 Alpha 2 standard. E.g. "GB" for the United Kingdom, "FR" for France, "US" for United States.
language	An IETF format language code (such as "es" for Spanish or "pt-BR" for Brazilian Portuguese). If no language is explicitly specified, we will look for an HTTP Accept-Language header like those sent by a browser and use the first language specified and if none are specified "en" (English) will be assumed.
limit	How many results should be returned (1-100). Default is 10.
min_confidence	An integer from 1-10. Only results with at least this confidence will be returned.
no_annotations	Logical (default FALSE), when TRUE the output will not contain annotations.
no_dedupe	Logical (default FALSE), when TRUE the output will not be deduplicated.

### Details

To get an API key to access OpenCage geocoding, register at <https://geocoder.opencagedata.com/pricing>. The free API key provides up to 2,500 calls a day. For ease of use, save your API key as an environment variable as described at [https://stat545-ubc.github.io/bit003\\_api-key-env-var.html](https://stat545-ubc.github.io/bit003_api-key-env-var.html). Both functions of the package will conveniently look for your API key using `Sys.getenv("OPENCAGE_KEY")` so if your API key is an environment variable called "OPENCAGE\_KEY" you don't need to input it manually.

The underlying data at OpenCage is updated about once a day. Note that the package uses 'memoise' with no timeout argument so that results are cached inside an active R session.

This function typically returns multiple results because of placename ambiguity; consider using the bounds parameter to limit the area searched.

**Value**

A list with

- results as a data.frame ('dplyr' 'tbl\_df') called results with one line per results,
- the number of results as an integer,
- the timestamp as a POSIXct object,
- rate\_info data.frame ('dplyr' 'tbl\_df') 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**

```
## Not run:
opencage_forward(placename = "Sarzeau")
opencage_forward(placename = "Islington, London")
opencage_forward(placename = "Triererstr 15,
                  Weimar 99423,
                  Deutschland")

## End(Not run)
```

---

opencage_reverse	<i>Reverse geocoding</i>
------------------	--------------------------

---

**Description**

Reverse geocoding, from latitude and longitude to placename(s).

**Usage**

```
opencage_reverse(latitude, longitude, key = opencage_key(), bounds = NULL,
                 countrycode = NULL, language = NULL, limit = 10,
                 min_confidence = NULL, no_annotations = NULL, no_dedupe = NULL)
```

**Arguments**

latitude	Latitude.
longitude	Longitude.
key	Your OpenCage key.
bounds	Provides the geocoder with a hint to the region that the query resides in. This value will restrict the possible results to the supplied region. The bounds parameter should be specified as 4 coordinate points forming the south-west and north-east corners of a bounding box. For example, bounds = c(-0.563160, 51.280430, 0.278970, 51.683)

(min long, min lat, max long, max lat).

countrycode	Restricts the results to the given country. The country code is a two letter code as defined by the ISO 3166-1 Alpha 2 standard. E.g. "GB" for the United Kingdom, "FR" for France, "US" for United States.
language	An IETF format language code (such as "es" for Spanish or "pt-BR" for Brazilian Portuguese). If no language is explicitly specified, we will look for an HTTP Accept-Language header like those sent by a browser and use the first language specified and if none are specified "en" (English) will be assumed.
limit	How many results should be returned (1-100). Default is 10.
min_confidence	An integer from 1-10. Only results with at least this confidence will be returned.
no_annotations	Logical (default FALSE), when TRUE the output will not contain annotations.
no_dedupe	Logical (default FALSE), when TRUE the output will not be deduplicated.

### Details

To get an API key to access OpenCage geocoding, register at <https://geocoder.opencagedata.com/pricing>. The free API key provides up to 2,500 calls a day. For ease of use, save your API key as an environment variable as described at [https://stat545-ubc.github.io/bit003\\_api-key-env-var.html](https://stat545-ubc.github.io/bit003_api-key-env-var.html). Both functions of the package will conveniently look for your API key using `Sys.getenv("OPENCAGE_KEY")` so if your API key is an environment variable called "OPENCAGE\_KEY" you don't need to input it manually.

The underlying data at OpenCage is updated about once a day. Note that the package uses 'memoise' with no timeout argument so that results are cached inside an active R session.

### Value

A list with

- results as a data.frame ('dplyr' 'tbl\_df') called results with one line per results,
- the number of results as an integer,
- the timestamp as a POSIXct object,
- rate\_info data.frame ('dplyr' 'tbl\_df') 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

```
## Not run:
opencage_reverse(latitude = 0, longitude = 0,
limit = 2)

## End(Not run)
```

# Index

## \*Topic **data**

code\_message, [2](#)

countrycodes, [2](#)

languagecodes, [2](#)

code\_message, [2](#)

countrycodes, [2](#)

languagecodes, [2](#)

opencage\_forward, [3](#)

opencage\_reverse, [4](#)