Package ‘opencage’

January 16, 2018

Type Package

Title Interface to the OpenCage API

Version 0.1.4

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/ropensci/opencage

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

Imports httr, jsonlite, dplyr, memoise

RoxygenNote 6.0.1

Suggests testthat, lintr, knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Maëlle Salmon [aut, cre],
Noam Ross [ctb],
Julia Silge [rev] (Julia Silge reviewed the package for rOpenSci, see https://github.com/ropensci/onboarding/issues/36.),
Jake Russ [ctb],
Daniel Possenriede [ctb]

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

Repository CRAN

Date/Publication 2018-01-16 11:31:26 UTC
R topics documented:

<table>
<thead>
<tr>
<th>R package</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>code_message</td>
<td>2</td>
</tr>
<tr>
<td>countrycodes</td>
<td>2</td>
</tr>
<tr>
<td>languagecodes</td>
<td>3</td>
</tr>
<tr>
<td>opencage_forward</td>
<td>3</td>
</tr>
<tr>
<td>opencage_reverse</td>
<td>5</td>
</tr>
</tbody>
</table>

Index

<table>
<thead>
<tr>
<th>code_message</th>
<th>API messages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

API messages

Format

Codes and corresponding messages from the API.

Examples

data("code_message")

<table>
<thead>
<tr>
<th>countrycodes</th>
<th>Country codes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

Country codes

Format

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

Examples

data("countrycodes")
languagecodes

<table>
<thead>
<tr>
<th>languagecodes</th>
<th>Language codes</th>
</tr>
</thead>
</table>

**Description**

Language codes

**Format**

All possible ISO 639-2 language codes.

**Examples**

data("languagecodes")

opencage_forward

**Forward geocoding**

**Description**

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

**Usage**

```r
opencage_forward(placename = NULL, 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**

- `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, 51.683000)` (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.

no_record
Logical (default FALSE), when TRUE no log entry of the query is created at OpenCage.

abbrv
Logical (default FALSE), when TRUE addresses are abbreviated (e.g. C. instead of Calle)

add_request
Logical (default TRUE), when FALSE the query text is removed from the results data frame.

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/bit03_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
```r
## Not run:
opencage_forward(placename = "Sarzeau")
opencage_forward(placename = "Islington, London")
opencage_forward(placename = "Triererstr 15,
                         Weimar 99423,
                         Deutschland")

## End(Not run)
```
Reverse geocoding, from latitude and longitude to placename(s).

Usage

```r
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, abbrev = FALSE, add_request = TRUE)
```

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.
- **no_record**: Logical (default FALSE), when TRUE no log entry of the query is created at OpenCage.
- **abbrev**: Logical (default FALSE), when TRUE addresses are abbreviated (e.g. C. instead of Calle)
- **add_request**: Logical (default TRUE), when FALSE the query text is removed from the results data frame.
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. 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 with be reset.

Examples

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

*Topic data
  code_message, 2
  countrycodes, 2
  languagecodes, 3

code_message, 2
countrycodes, 2
languagecodes, 3

opencage_forward, 3
opencage_reverse, 5