Package 'gmapsdistance'

August 28, 2018

Type Package

Title Distance and Travel Time Between Two Points from Google Maps

Version 3.4

Date 2018-08-28

Author Rodrigo Azuero Melo & & Demetrio Rodriguez T & David Zarruk

URL <https://github.com/rodazuero/gmapsdistance>

Maintainer Rodrigo Azuero Melo <rodazuero@gmail.com>

Description Get distance and travel time between two points from Google Maps.
Four possible modes of transportation (bicycling, walking, driving and
public transportation).

License GPL (>= 2)

Imports RCurl, XML, methods, stats

RoxygenNote 6.1.0

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2018-08-28 17:22:15 UTC

R topics documented:

---

get.api.key　　　　　　　*Get the Google Maps API key*

Description

This function returns the user's Google Maps API key that was defined with `set.api.key`.

Usage

```
get.api.key()
```

Value

the user's api key

Examples

```
get.api.key()
```

---

gmapsdistance　　　　　　*gmapsdistance*

Description

Compute Distance with Google Maps

Usage

```
gmapsdistance(
    origin,
    destination,
    combinations = "all",
    mode,
    key = get.api.key(),
    shape = "wide",
    avoid = "",
    departure = "now",
    dep_date = "",
    dep_time = "",
    traffic_model = "None",
    arrival = "",
    arr_date = "",
    arr_time = ""
)
```

Arguments

origin	A string or vector of strings containing the description of the starting point(s). Should be inside of quoutes (""). If more than one word for a same location is used, they should be separated by a plus sign e.g. "Bogota+Colombia". Coordinates in LAT-LONG format are also a valid input as long as they can be identified by Google Maps.
destination	A string or vector of strings containing the description of the end point(s). Should be the same format as the variable "origin".
combinations	When the origin and destination entries are vectors, the user can specify if the function computes all possible combinations between origins and destinations, or only pairwise distance and times. Should be inside of double quotes (",") and one of the following: "all", "pairwise".
	If the combinations is set to "pairwise", the origin and destination vectors must have the same lenght.
mode	A string containing the mode of transportation desired. Should be inside of double quotes (",") and one of the following: "bicycling", "walking", "transit" or "driving".
key	In order to use the Google Maps Distance Matrix API it is necessary to have an API key. The key should be inside of quotes. Example: "THISISMYKEY". This key an also be set using `set.api.key("THISISMYKEY")`.
shape	A string that specifies the shape of the distance and time matrices to be returned. Should be inside of double quotes (",") and one of the following: "long" or "wide".
	If the function is used to find the distance/time for one origin and one destination, the shape does not matter. If there is more than one city as origin or destination, "long" will return a matrix in long format and "wide" will return a matrix in wide format. The shape is set as wide be default.
avoid	When the mode is set to "driving", the user can find the time and distance of the route by avoiding tolls, highways, indoor and ferries. Should be inside of double quotes (",") and one of the following: "tolls", "highways", "ferries", "indoor".
	ONLY works with a Google Maps API key.
departure	The time and distance can be comptued at the desired time of departure. The option departure is the number of seconds since January 1, 1970 00:00:00 UCT. Alternatively, the user can use the dep_date and dep_time options to set the departure date and time.
	If no value is set for departure, dep_date and dep_time, the departure time is set to the present.
	ONLY works with a Google Maps API key AND MUST be according to UCT time.
dep_date	Instead of using the departure option, the user can set the departure date and time using dep_date and dep_time options.
	If no value is set for departure, dep_date and dep_time, the departure time is set to the present.
	ONLY works with a Google Maps API key AND MUST be according to UCT time.

dep_time	Instead of using the departure option, the user can set the departure date and time using dep_date and dep_time options.
	If no value is set for departure, dep_date and dep_time, the departure time is set to the present.
	ONLY works with a Google Maps API key AND MUST be according to UCT time.
traffic_model	When the mode is set to "driving", the user can find the times and distances using different traffic models. Should be a string and one of the following: "optimistic", "pessimistic", "best_guess" or "None" (default).
	ONLY works with a Google Maps API key and with a departure time.
arrival	The time and distance can be comptued to arrive at a predetermined time. The option arrival is the number of seconds since January 1, 1970 00:00:00 UCT. Alternatively, the user can use the arr_date and arr_time options to set the arrival date and time.
	The user cannot input both departure and arrival times.
	ONLY works with a Google Maps API key AND MUST be according to UCT time.
arr_date	Instead of using the arrival option, the user can set the arrival date and time using arr_date and arr_time options.
	The user cannot input both departure and arrival times.
	ONLY works with a Google Maps API key AND MUST be according to UCT time.
arr_time	Instead of using the arrival option, the user can set the arrival date and time using arr_date and arr_time options.
	The user cannot input both departure and arrival times.
	ONLY works with a Google Maps API key AND MUST be according to UCT time.

Details

The function gmapsdistance uses the Google Maps Distance Matrix API in order to compute the distance(s) and time(s) between two points. In order to be able to use the function you will need an API key and enable the Distance Matrix API in the Google Developers Console For more information about how to get a key, go to https://developers.google.com/maps/documentation/distance-matrix/get-api-key#key For more information about the Google Maps Distance Matrix API go to https://developers.google.com/maps/documentation/distance-matrix/intro?hl=en

Value

A list with the traveling time(s) and distance(s) between origin(s) and destination(s) and the status #' @examples # Example 1 results = gmapsdistance(origin = "Washington+DC", destination = "New+York+City+NY", mode = "driving") results

Example 2 results = gmapsdistance(origin = "38.1621328+24.0029257", destination = "37.9908372+23.7383394", mode = "walking") results

Example 3 results = gmapsdistance(origin = c("Seattle+WA", "Miami+FL"), destination = c("Chicago+IL", "Philadelphia+PA"), mode = "bicycling", dep_date = "2022-08-16", dep_time = "20:40:00")

results

Example 4 origin = c("Washington+DC", "Miami+FL") destination = c("Los+Angeles+CA", "Austin+TX", "Chicago+IL") results = gmapsdistance(origin, destination, mode = "driving", shape = "long") results

Example 5 origin = c("40.431478+-80.0505401", "33.7678359+-84.4906438") destination = c("43.0995629+-79.0437609", "41.7096483+-86.9093986") results = gmapsdistance(origin, destination, mode = "bicycling", shape="long") results

Example 6 (do not run – needs an API key) # results = gmapsdistance(origin = c("Washington+DC", "New+York+NY"), # destination = c("Los+Angeles+CA", "Austin+TX"), # mode = "driving", # departure = 1514742000, # traffic_model = "pessimistic", # shape = "long", # key=APIkey) # # results

EXAMPLE 7 (do not run – needs an API key) # results = gmapsdistance(origin = c("Washington+DC", "New+York+NY"), # destination = c("Los+Angeles+CA", "Austin+TX"), # mode = "driving", # avoid = "tolls", # key=APIkey) # # results

---

pkg.env	*Define package environment*

Description

pkg.env is a package environment that contains the variable api.key with the user's Google Maps API key

Usage

pkg.env

Format

An object of class environment of length 1.

---

set.api.key	*Set the Google Maps API key*

Description

This function stores a user's Google Maps API key as the package's environmental variable

Usage

set.api.key(key)

Arguments

key   is the user's Google Maps API key

Examples

```
#DONTRUN
set.api.key("MY-GOOGLE-MAPS-API-KEY")
```

Index