Package ‘rflights’

September 18, 2019

Type Package
Title Query Plane Tickets using the 'Kiwi' API
Version 0.1.103
Maintainer Juan Cruz Rodriguez <jcrodriguez@unc.edu.ar>
Description Query plane tickets, from several airlines, using the 'Kiwi' API (similar to 'Google Flights').
   The API is documented at <https://docs.kiwi.com/>.
License GPL-3
BugReports https://github.com/jcrodriguez1989/rflights/issues
Encoding UTF-8
LazyData true
Depends R (>= 2.10)
Imports httr, methods
Suggests testthat, covr, knitr, rmarkdown
VignetteBuilder knitr
RoxygenNote 6.1.1
NeedsCompilation no
Author Juan Cruz Rodriguez [aut, cre]
Repository CRAN
Date/Publication 2019-09-18 21:20:02 UTC

R topics documented:

afrika ................................................................. 2
asia ................................................................. 2
country_code ...................................................... 3
europe ............................................................ 3
find_location ..................................................... 4
Description

For each continent, a character vector containing its countries codes.

Usage

africa

descriptions

Format

A data frame with three variables:

africa codes of countries in Africa
asia codes of countries in Asia
europe codes of countries in Europe
north_america codes of countries in North America
oceania codes of countries in Oceania
south_america codes of countries in South America

Description

For each continent, a character vector containing its countries codes.

Usage

asia
**country_code**

**Format**

A data frame with three variables:

- **africa** codes of countries in Africa
- **asia** codes of countries in Asia
- **europe** codes of countries in Europe
- **north_america** codes of countries in North America
- **oceania** codes of countries in Oceania
- **south_america** codes of countries in South America

<table>
<thead>
<tr>
<th>country_code</th>
<th>Country code.</th>
</tr>
</thead>
</table>

**Description**

Contains the name, ID code, and continent of each country.

**Usage**

country_code

**Format**

A data frame with three variables:

- **name** country name
- **code** country code to use as fly_from and fly_to
- **continent** continent name

<table>
<thead>
<tr>
<th>europe</th>
<th>Continents' countries.</th>
</tr>
</thead>
</table>

**Description**

For each continent, a character vector containing its countries codes.

**Usage**

europe
find_location

Format

A data frame with three variables:

africa codes of countries in Africa
asia codes of countries in Asia
europe codes of countries in Europe
north_america codes of countries in North America
oceania codes of countries in Oceania
south_america codes of countries in South America

Description

Get location ID depending on a query term using the [Kiwi API](https://docs.kiwi.com/).

Usage

find_location(term, location_types = NA, locale = "en-US")

Arguments

term searched term (for suggestions). This parameter expects a full IATA code. If IATA code is not given, the search will go through other available fields: 'name' or 'code' of the location. It also depends on the 'location_types' specified eg. airport, city, country. The search that is used behind the scenes is elasticsearch. It returns data based on relevancy and many other factors.

location_types list of desired location output, accepted values: station, airport, bus_station, city, autonomous_territory, subdivision, country, region, continent.

locale desired locale output - this is the language of the results. Should any other locale be used other than the specified locales.

Examples

cba_locs <- find_location("Cordoba", location_types = c("city", "airport"))
# show some info of the found locations
lapply(cba_locs, function(act_loc) {
  c(act_loc$name, act_loc$country$name)
})
**get_flights**

Query flight prices.

**Description**

Query flight prices using the [Kiwi API](https://docs.kiwi.com/).

**Usage**

```r
get_flights(fly_from, fly_to = "anywhere", date_from = Sys.Date(),
            date_to = date_from + 1, return_from = NA, return_to = NA,
            curr = "USD", price_from = NA, price_to = NA,
            other_params = list())
```

**Arguments**

- **fly_from**: ID of the departure location. It accepts multiple values separated by comma, these values might be airport codes, city IDs, two letter country codes, metropolitan codes and radiuses as well as subdivision, region, autonomous_territory, continent and specials (Points of interest, such as Times Square). Some locations have the same code for airport and metropolis (city), e.g. DUS stands for metro code Duesseldorf, Moenchengladbach and Weeze as well as Duesseldorf airport. See the following examples: fly_from=city:DUS will match all airports in "DUS", "MGL" and "NRN" (all in the city of Duesseldorf) fly_from=DUS will do the same as the above fly_from=airport:DUS will only match airport "DUS" Radius needs to be in form lat-lon-xkm. The number of decimal places for radius is limited to 6. E.g. -23.24-47.86-500km for places around Sao Paulo. 'LON' - checks every airport in London, 'LHR' - checks flights from London Heathrow, 'UK' - flights from the United Kingdom. Link to Locations API.

- **fly_to**: ID of the arrival destination. It accepts the same values in the same format as the fly_from parameter.

- **date_from**: search flights from this date (dd/mm/YYYY). Use parameters date_from and date_to as a date range for the flight departure. Parameters 'date_from=01/05/2016' and 'date_to=30/05/2016' mean that the departure can be anytime between the specified dates. For the dates of the return flights, use the 'return_to' and 'return_from' or 'nights_in_dst_from' and 'nights_in_dst_to' parameters.

- **date_to**: search flights upto this date (dd/mm/YYYY).

- **return_from**: min return date of the whole trip (dd/mm/YYYY).

- **return_to**: max return date of the whole trip (dd/mm/YYYY).

- **curr**: use this parameter to change the currency in the response.

- **price_from**: result filter, minimal price

- **price_to**: result filter, maximal price

- **other_params**: named list of other params from https://docs.kiwi.com/#flights-flights-get
Examples

# get Argentina and toulouse IDs
arg_id <- find_location("Argentina", "country")[[1]]$id # AR
tl_id <- find_location("toulouse", "city")[[1]]$id

# get flights with no specified date
flights <- get_flights(arg_id, tl_id)
sapply(flights, function(x) x$price)

north_america  Continents’ countries.

Description
For each continent, a character vector containing its countries codes.

Usage
north_america

Format
A data frame with three variables:
africa  codes of countries in Africa
asia   codes of countries in Asia
europe  codes of countries in Europe
north_america  codes of countries in North America
oceania  codes of countries in Oceania
south_america  codes of countries in South America

oceania  Continents’ countries.

Description
For each continent, a character vector containing its countries codes.

Usage
oceania
Format

A data frame with three variables:

- africa codes of countries in Africa
- asia codes of countries in Asia
- europe codes of countries in Europe
- north_america codes of countries in North America
- oceania codes of countries in Oceania
- south_america codes of countries in South America

---

**Description**

For each continent, a character vector containing its countries codes.

**Usage**

south_america

**Format**

A data frame with three variables:

- africa codes of countries in Africa
- asia codes of countries in Asia
- europe codes of countries in Europe
- north_america codes of countries in North America
- oceania codes of countries in Oceania
- south_america codes of countries in South America
Index

*Topic datasets
  africa, 2
  asia, 2
  country_code, 3
  europe, 3
  north_amERICA, 6
  oceania, 6
  south_america, 7
  africa, 2
  asia, 2
  country_code, 3
  europe, 3
  find_location, 4
  get_flights, 5
  north_america, 6
  oceania, 6
  south_america, 7