

Package ‘cyclestreets’

May 3, 2018

Type Package

Title Cycle Routing and Data for Cycling Advocacy

Version 0.1.5

Description An interface to the cycle routing/data services provided by 'CycleStreets', a not-for-profit social enterprise and advocacy organisation. The application programming interfaces (APIs) provided by 'CycleStreets' are documented at (<<https://www.cyclestreets.net/api/>>). The focus of this package is the journey planning API, which aims to emulate the routes taken by a knowledgeable cyclist. An innovative feature of the routing service of its provision of fastest, quietest and balanced profiles. These represent routes taken to minimise time, avoid traffic and compromise between the two, respectively.

License GPL-3

Encoding UTF-8

LazyData true

Imports sf, magrittr, jsonlite, httr, stringr

RoxygenNote 6.0.1

NeedsCompilation no

Author Robin Lovelace [aut, cre] (<<https://orcid.org/0000-0001-5679-6536>>),
Martin Lucas-Smith [aut],
Eric Krueger [ctb]

Maintainer Robin Lovelace <rob00x@gmail.com>

Repository CRAN

Date/Publication 2018-05-03 12:28:11 UTC

R topics documented:

journey	2
json2sf_cs	3

Index	4
--------------	----------

 journey

Plan a journey with CycleStreets.net

Description

R interface to the CycleStreets.net journey planning API, a route planner made by cyclists for cyclists. See cyclestreets.net/api for details.

Usage

```
journey(from, to, plan = "fastest", silent = TRUE, pat = NULL,
        base_url = "https://www.cyclestreets.net", reporterrors = TRUE,
        save_raw = "FALSE", cols = c("name", "distances", "time", "busynance",
        "elevations", "start_longitude", "start_latitude", "finish_longitude",
        "finish_latitude"))
```

Arguments

from	Longitude/Latitude pair, e.g. c(-1.55, 53.80)
to	Longitude/Latitude pair, e.g. c(-1.55, 53.80)
plan	Text strong of either "fastest" (default), "quietest" or "balanced"
silent	Logical (default is FALSE). TRUE hides request sent.
pat	The API key used. By default this uses <code>Sys.getenv("CYCLESTREETS")</code> .
base_url	The base url from which to construct API requests (with default set to main server)
reporterrors	Boolean value (TRUE/FALSE) indicating if cyclestreets (TRUE by default) should report errors (FALSE by default).
save_raw	Boolean value which returns raw list from the json if TRUE (FALSE by default).
cols	Columns to be included in the result, a character vector or NULL for all available columns (see details for default)

Details

Requires the internet and a CycleStreets.net API key. CycleStreets.net does not yet work worldwide.

You need to have an api key for this code to run. By default it uses the CYCLESTREETS environment variable. This can be set with `usethis::edit_r_environ()`.

A full list of variables (cols) available is represented by:

```
c("time", "busynance", "signalledJunctions", "signalledCrossings",
  "name", "walk", "elevations", "distances", "start", "finish",
  "startSpeed", "start_longitude", "start_latitude", "finish_longitude",
  "finish_latitude", "crow_fly_distance", "event", "whence", "speed",
  "itinerary", "clientRouteId", "plan", "note", "length", "quietness",
  "west", "south", "east", "north", "leaving", "arriving", "grammesCO2saved",
  "calories", "edition", "geometry")
```

See Also

json2sf_cs

Examples

```
## Not run:
from = c(-1.55, 53.80) # geo_code("leeds")
to = c(-1.76, 53.80) # geo_code("bradford uk")
r1 = journey(from, to)
sf:::plot.sf(r1)
to = c(-2, 53.5) # towards manchester
r1 = journey(from, to)
r2 = journey(from, to, plan = "balanced")
plot(r1["busynance"], reset = FALSE)
plot(r2["busynance"], add = TRUE)
r3 = journey(from, to, silent = FALSE)
r4 = journey(from, to, save_raw = TRUE)
r5 = journey(from, to, cols = NULL)

## End(Not run)
```

json2sf_cs

Convert output from CycleStreets.net into sf object

Description

Convert output from CycleStreets.net into sf object

Usage

```
json2sf_cs(obj, cols = NULL)
```

Arguments

obj	Object from CycleStreets.net read-in with
cols	Columns to be included in the result, a character vector or NULL for all available columns (see details for default)

Examples

```
from = "Leeds Rail Station"
to = "University of Leeds"
# save result from the API call to journey.json
# res_json = stplanr::route_cyclestreet(from, to, silent = FALSE, save_raw = TRUE)
# jsonlite::write_json(res_json, "inst/extdata/journey.json")
f = system.file(package = "cyclestreets", "extdata/journey.json")
obj = jsonlite::read_json(f, simplifyVector = TRUE)
rsf = json2sf_cs(obj)
sf:::plot.sf(rsf)
json2sf_cs(obj, cols = c("time", "busynance", "elevations"))
```

Index

journey, [2](#)
json2sf_cs, [3](#)