

# Package ‘cyclestreets’

September 26, 2020

**Type** Package

**Title** Cycle Routing and Data for Cycling Advocacy

**Version** 0.3.0

**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 is 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, geodist

**RoxygenNote** 7.1.1

**Depends** R (>= 3.6.0)

**Suggests** stplanr, covr

**NeedsCompilation** no

**Author** Robin Lovelace [aut, cre] (<<https://orcid.org/0000-0001-5679-6536>>),  
Martin Lucas-Smith [aut],  
Eric Krueger [ctb],  
Joey Talbot [aut] (<<https://orcid.org/0000-0002-6520-4560>>)

**Maintainer** Robin Lovelace <[rob00x@gmail.com](mailto:rob00x@gmail.com)>

**Repository** CRAN

**Date/Publication** 2020-09-26 04:20:06 UTC

## R topics documented:

cyclestreets_column_names	2
journey	2
json2sf_cs	4
smooth_with_cutoffs	6

<b>Index</b>	<b>7</b>
--------------	----------

---

cyclestreets_column_names	<i>Prices of 50,000 round cut diamonds.</i>
---------------------------	---

---

### Description

Variables provided by CycleStreets in their journey data

### Usage

```
cyclestreets_column_names
```

### Format

An object of class character of length 44.

### Source

<https://www.cyclestreets.net/>

---

journey	<i>Plan a journey with CycleStreets.net</i>
---------	---

---

### Description

R interface to the CycleStreets.net journey planning API, a route planner made by cyclists for cyclists. See [cyclestreets.net/api](https://www.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"),
    cols_extra = c("crow_fly_distance", "event", "whence", "speed", "itinerary",
                  "clientRouteId", "plan", "note", "length", "quietness", "west", "south", "east",
                  "north", "leaving", "arriving", "grammesCO2saved", "calories", "edition",
                  "gradient_segment", "elevation_change", "provisionName", "quietness_segment"),
    smooth_gradient = TRUE,
    distance_cutoff = 50,
    gradient_cutoff = 0.1,
    n = 3
)

```

### 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)
cols_extra	Additional columns to be added providing summaries of gradient and other variables
smooth_gradient	Identify and fix anomalous gradients? TRUE by default. See <a href="https://github.com/Robinlovelace/cyclestreet">https://github.com/Robinlovelace/cyclestreet</a>
distance_cutoff	Distance (m) used to identify anomalous gradients
gradient_cutoff	Gradient (% , e.g. 0.1 being 10%) used to identify anomalous gradients
n	The number of segments to use to smooth anomalous gradients. The default is 3, meaning segments directly before, after and including the offending segment.

### 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)
names(r1)
r1[1:2, ]
r1$grammesCO2saved
r1$calories
plot(r1[1:4])
plot(r1[10:ncol(r1)])
to = c(-2, 53.5) # towards Manchester
r1 = journey(from, to)
names(r1)
r2 = journey(from, to, plan = "balanced")
plot(r1["quietness_segment"], reset = FALSE)
plot(r2["quietness_segment"], add = TRUE)
r3 = journey(from, to, silent = FALSE)
r4 = journey(from, to, save_raw = TRUE)
r5 = journey(c(-1.524, 53.819), c(-1.556, 53.806))
plot(r5["gradient_segment"])
plot(r5["gradient_smooth"])

## 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,
    cols_extra = c("elevation_start", "elevation_end", "gradient_segment",
                  "elevation_change", "provisionName", "quietness_segment"),
    smooth_gradient = FALSE,
    distance_cutoff = 50,
    gradient_cutoff = 0.1,
    n = 3
  )

```

## 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)
cols_extra	Additional columns to be added providing summaries of gradient and other variables
smooth_gradient	Identify and fix anomalous gradients? TRUE by default. See <a href="https://github.com/Robinlovelace/cyclestreets">https://github.com/Robinlovelace/cyclestreets</a>
distance_cutoff	Distance (m) used to identify anomalous gradients
gradient_cutoff	Gradient (% , e.g. 0.1 being 10%) used to identify anomalous gradients
n	The number of segments to use to smooth anomalous gradients. The default is 3, meaning segments directly before, after and including the offending segment.

## Examples

```

from = "Leeds Rail Station"
to = "University of Leeds"
# from_point = tmertools::geocode_OSM(from)
# to_point = tmertools::geocode_OSM(to)
from_point = c(-1.54408, 53.79360)
to_point = c(-1.54802, 53.79618)
# save result from the API call to journey.json
# res_json = journey(from_point, to_point, 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, cols = c("distances"))
names(rsf)
rsf
rsf2 = json2sf_cs(obj, cols = NULL, cols_extra = NULL)
names(rsf2)
# stplanr::line2points(rsf) extract start and end points
sf::plot.sf(rsf)
json2sf_cs(obj, cols = c("time", "busynance", "elevations"))
json2sf_cs(obj, cols = c("distances"), smooth_gradient = TRUE,
            gradient_cutoff = 0.05, distance_cutoff = 50)

```

---

smooth\_with\_cutoffs    *Identify and smooth-out anomalous gradient values*

---

### Description

When distance\_cutoff and gradient\_cutoff thresholds are both broken for route segments, this function treats them as anomalous and sets the offending gradient values to the mean of the n segments closest to (in front of and behind) the offending segment.

### Usage

```
smooth_with_cutoffs(
  gradient_segment,
  elevation_change,
  distances,
  distance_cutoff = 50,
  gradient_cutoff = 0.1,
  n = 3
)
```

### Arguments

gradient_segment	The gradient for each segment from CycleStreets.net
elevation_change	The difference between the maximum and minimum elevations within each segment
distances	The distance of each segment
distance_cutoff	Distance (m) used to identify anomalous gradients
gradient_cutoff	Gradient (% , e.g. 0.1 being 10%) used to identify anomalous gradients
n	The number of segments to use to smooth anomalous gradients. The default is 3, meaning segments directly before, after and including the offending segment.

### Examples

```
f = system.file(package = "cyclestreets", "extdata/journey.json")
obj = jsonlite::read_json(f, simplifyVector = TRUE)
rsf = json2sf_cs(obj, cols = c("distances"))
rsf$gradient_segment
rsf$elevation_change
rsf$distances
smooth_with_cutoffs(rsf$gradient_segment, rsf$elevation_change, rsf$distances)
smooth_with_cutoffs(rsf$gradient_segment, rsf$elevation_change, rsf$distances, 20, 0.05)
smooth_with_cutoffs(rsf$gradient_segment, rsf$elevation_change, rsf$distances, 200, 0.02)
smooth_with_cutoffs(rsf$gradient_segment, rsf$elevation_change, rsf$distances, 200, 0.02, n = 5)
```

# Index

## \* **datasets**

[cylestreets\\_column\\_names](#), [2](#)

[cylestreets\\_column\\_names](#), [2](#)

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

[json2sf\\_cs](#), [4](#)

[smooth\\_with\\_cutoffs](#), [6](#)