Package 'walkalytics'

April 23, 2018

Type Package

Title Interface to the 'Walkalytics' API for Calculating Walking Isochrones

Version 0.1.0

Description
A wrapper for the 'walkalytics' API (see <https://dev.walkalytics.com/>), which is a web service that calculates walking isochrones for source locations. Basic, limited API access is free.

License GPL (>= 2)

Encoding UTF-8

LazyData true

URL https://dev.walkalytics.com

BugReports https://github.com/zumbov2/walkalytics/issues

RoxygenNote 6.0.1

Imports httr, dplyr, stringr, purrr, tibble, base64enc, sp

NeedsCompilation no

Author David Zumbach [aut, cre]

Maintainer David Zumbach <david.zumbach@gfzb.ch>

Repository CRAN

Date/Publication 2018-04-23 11:19:05 UTC

R topics documented:

esri_to_sgdf 2
get_stops 2
isochrone 3
isochrone_esri 5
isochrone_png 6
isochrone_pois 7
pixel_walktimes 8
pois_walktimes 9
pubtrans_ch_nearby 10
save_png 11

Index **12**

`esri_to_sgdf` *Convert a base64-encoded gzipped Esri ASCII grid to an object of class SpatialGridDataFrame*

Description

`esri_to_sgdf` converts a response object from a isochrone_esri call to the `walkalytics` isochrone API to an object of class SpatialGridDataFrame-class.

Usage

```
esri_to_sgdf(isochrone_esri)
```

Arguments

`isochrone_esri`	a response object from a isochrone_esri call to the `walkalytics` isochrone API.

Examples

```
isochrone_esri(x = 895815, y = 6004839, key = "abcd1234") %>% esri_to_sgdf()
```

`get_stops` *Extract walking times to nearby public transportation stops*

Description

`get_stops` processses a response object from a pubtrans_ch_nearby call to the `walkalytics` pubtrans API. Returns the nearby public tansportation stops, ordered by walking time.

Usage

```
get_stops(pubtrans_ch_nearby)
```

Arguments

`pubtrans_ch_nearby`	a response object from a pubtrans_ch_nearby call to the `walkalytics` pubtrans API.

Details

To get an API key, you need to register at `https://dev.walkalytics.com/signin`. With the free starter account, you can make up to 100 calls a week to the API.

Value

A `data.frame` (`tibble::tibble`) that contains:

- `name` name of the public tansportation stop (station).
- `walktime` estimated walking time from the starting point to the station in minutes.
- `station_category` category of the station, ranging from high to low service frequency (1 to 5; unassigned >= 90).
- `latitude` latitude of the station.
- `longitude` longitude of the station.
- `coordinates_type` type of geodetic system.
- `transport_category` type of station.
- `id` official ID of the station.

References

Walkalytics API documentations

Examples

```
pubtrans_ch_nearby(x = 8.05, y = 47.3, key = "abcd1234") %>% get_stops()
```

`isochrone` *Issue a walkalytics isochrone query*

Description

`isochrone` calls the `walkalytics` isochrone API which calculates the walking isochrone for a source location. The repsonse object contains a base64-encoded raster file, containing 4 classes (as PNG file, this is default) or the actual travel times in seconds for every pixel (as a gzipped Esri ASCII grid). If a set of points-of-interest (POIs) is given, the duration time for walking from the source location to each POI is calculated.

Usage

```
isochrone(x, y, epsg = 3857, max_min = 1000, raw_data = FALSE,
  pois = NULL, only_pois = FALSE, break_values = c(0, 3, 6, 9, 13),
  key = "my_walkalytics_key")
```

Arguments

x	x-coordinate of the source location (coordinate system is WGS/84 Pseudo Mercator).
y	y-coordinate of the source location (coordinate system is WGS/84 Pseudo Mercator).
epsg	EPSG code for coordinate system of the x- and y-coordinate.
max_min	maximum number of minutes for the isochrone.
raw_data	if TRUE, the API returns a gzipped Esri ASCII grid with traveltimes for every pixel. If FALSE, it returns a PNG file with classified isochrones.
pois	a data.frame to specify a set of points-of-interest (POIs). The API calculates the duration time for walking from the source location to each POI. The following columns are required: • x x-coordinate of the source location (EPSG:3857). • y y-coordinate of the source location (EPSG:3857). • id name of POI (optional)
only_pois	if TRUE, the API only returns an annotated list of the points-of-interest (POIs). No isochrone raster will be included in the response.
break_values	a vector of break values (walking time in minutes) for the classification of the PNG result.
key	your walkalytics subscription key which provides access to the API.

Details

To get an API key, you need to register at https://dev.walkalytics.com/signin. With the free starter account, you can make up to 100 calls a week to the API.

Value

The response object from the request. Use esri_to_sgdf, pixel_walktimes, save_png, or pois_walktimes to process the response.

References

Walkalytics API documentations

Examples

```
isochrone(x = 895815, y = 6004839, key = "abcd1234")
```

`isochrone_esri` *Issue a walkalytics isochrone query*

Description

`isochrone_esri` calls the `walkalytics` isochrone API which calculates the walking isochrone for a source location and returns a repsonse object that contains a base64-encoded gzipped Esri ASCII grid with walking times for every pixel.

Usage

```
isochrone_esri(x, y, epsg = 3857, max_min = 1000,
  key = "my_walkalytics_key")
```

Arguments

`x`	x-coordinate of the source location (coordinate system is WGS/84 Pseudo Mercator).
`y`	y-coordinate of the source location (coordinate system is WGS/84 Pseudo Mercator).
`epsg`	EPSG code for coordinate system of the x- and y-coordinate.
`max_min`	maximum number of minutes for the isochrone.
`key`	your `walkalytics` subscription key which provides access to the API.

Details

To get an API key, you need to register at `https://dev.walkalytics.com/signin`. With the free starter account, you can make up to 100 calls a week to the API.

Value

The response object from the request. Use esri_to_sgdf to convert the base64-encoded gzipped Esri ASCII grid to an object of class SpatialGridDataFrame-class. Use pixel_walktimes to directly extract walking times for every pixel.

References

Walkalytics API documentations

Examples

```
isochrone_esri(x = 895815, y = 6004839, key = "abcd1234")
```

isochrone_png *Issue a walkalytics isochrone query*

Description

isochrone_png calls the walkalytics isochrone API which calculates the walking isochrone for a source location and returns a repsonse object that contains a base64-encoded raster as PNG file with classified isochrones.

Usage

```
isochrone_png(x, y, epsg = 3857, max_min = 1000, break_values = c(0, 3, 6,
  9, 13), key = "my_walkalytics_key")
```

Arguments

x	x-coordinate of the source location (coordinate system is WGS/84 Pseudo Mercator).
y	y-coordinate of the source location (coordinate system is WGS/84 Pseudo Mercator).
epsg	EPSG code for coordinate system of the x- and y-coordinate.
max_min	integer. Maximum number of minutes for the isochrone.
break_values	a vector of break values (walking time in minutes) for the classification of the PNG result.
key	your walkalytics subscription key which provides access to the API.

Details

To get an API key, you need to register at https://dev.walkalytics.com/signin. With the free starter account, you can make up to 100 calls a week to the API.

Value

The response object from the request. Use save_png to save the base64-encoded PNG to file.

References

Walkalytics API documentations

Examples

```
isochrone_png(x = 895815, y = 6004839, key = "abcd1234")
```

`isochrone_pois` *Issue a walkalytics isochrone query*

Description

`isochrone_pois` calls the `walkalytics` isochrone API which calculates the walking times from a source location and to a given set of points-of-interest (POIs).

Usage

```
isochrone_pois(x, y, epsg = 3857, max_min = 1000, pois,
  key = "my_walkalytics_key")
```

Arguments

`x`	x-coordinate of the source location (coordinate system is WGS/84 Pseudo Mercator).
`y`	y-coordinate of the source location (coordinate system is WGS/84 Pseudo Mercator).
`epsg`	EPSG code for coordinate system of the x- and y-coordinate.
`max_min`	maximum number of minutes for the isochrone.
`pois`	a `data.frame` to specify a set of points-of-interest (POIs). The API calculates the duration time for walking from the source location to each POI. The following columns are required: • `x` x-coordinate of the source location (EPSG:3857). • `y` y-coordinate of the source location (EPSG:3857). • `id` name of POI (optional)
`key`	your `walkalytics` subscription key which provides access to the API.

Details

To get an API key, you need to register at `https://dev.walkalytics.com/signin`. With the free starter account, you can make up to 100 calls a week to the API.

Value

The response object from the request. Use pois_walktimes to directly extract the walking times between the source location and the points-of-interest.

References

Walkalytics API documentations

Examples

```
# Generate set of POIs
x <- c(895777, 896044, 895639)
y <- c(6004833, 6004886, 6005147)
id <- c("pupil1", "pupil2", "pupil3")
pupils <- data.frame(x = x, y = y, id = id)

# Issue query
isochrone_pois(x = 895815, y = 6004839, pois = pupils, key = "abcd1234")
```

`pixel_walktimes` *Extract pixel-accurate walking times from walkalytics raw data isochrones*

Description

`pixel_walktimes` extracts walking times for every pixel from a response object from a isochrone_esri call.

Usage

```
pixel_walktimes(isochrone_esri)
```

Arguments

`isochrone_esri`	a response object from a isochrone_esri call to the `walkalytics` isochrone API.

Value

A `data.frame` (`tibble::tibble`) that contains:

- `walktime` estimated walking times in seconds from the starting point to every pixel.
- `x` x-coordinate of the pixel.
- `y` y-coordinate of the pixel.

Examples

```
isochrone_esri(x = 896488, y = 6006502, key = "abcd1234") %>% pixel_walktimes()
```

pois_walktimes *Extract walking times to points-of-interest*

Description

pois_walktimes processses a response object from a isochrone_pois call to the walkalytics pub-trans API. Returns walking times from the source location to the given points-of-interest, ordered by walking time.

Usage

```
pois_walktimes(isochrone_pois)
```

Arguments

isochrone_pois	a response object from a isochrone_pois call to the walkalytics isochrone API.

Details

To get an API key, you need to register at https://dev.walkalytics.com/signin. With the free starter account, you can make up to 100 calls a week to the API.

Value

A data.frame (tibble::tibble) that contains:

- id id of the point-of-interest.
- walktime estimated walking time from the starting point to the point-of-interest in seconds.
- x x-coordinate of the point-of-interest.
- y y-coordinate of the point-of-interest.

References

Walkalytics API documentations

Examples

```
# Generate set of POIs
x <- c(895777, 896044, 895639)
y <- c(6004833, 6004886, 6005147)
id <- c("pupil1", "pupil2", "pupil3")
pupils <- data.frame(x = x, y = y, id = id)

# Issue query
isochrone_pois(x = 895815, y = 6004839, pois = pupils, key = "abcd1234") %>% pois_walktimes()
```

`pubtrans_ch_nearby` *Issue a walkalytics pubtrans query*

Description

`pubtrans_ch_nearby` calls the `walkalytics` pubtrans API which returns nearby public tansportation stops for Switzerland.

Usage

```
pubtrans_ch_nearby(x, y, max_walktime = 10, key = "my_walkalytics_key")
```

Arguments

`x`	latitude of starting point.
`y`	longitude of starting point.
`max_walktime`	output filter for maximum walking time in minutes.
`key`	your `walkalytics` subscription key which provides access to the API.

Details

To get an API key, you need to register at `https://dev.walkalytics.com/signin`. With the free starter account, you can make up to 100 calls a week to the API.

Value

The response object from the request. Use get_stops to directly extract nearby public transportation stops and estimated walking times from the starting point to the stations.

References

Walkalytics API documentations

Examples

```
pubtrans_ch_nearby(x = 8.0526331, y = 47.3933375, max_walktime = 10, key = "abcd1234")
```

save_png *Save a base64-encoded PNG to file*

Description

save_png decodes a base64-encoded PNG of a response object from a isochrone_png call and saves it to file.

Usage

```
save_png(isochrone_png, file = "isochrone.png")
```

Arguments

isochrone_png	a response object from a isochrone_png call to the walkalytics isochrone API.
file	character vector, containing file name or path

Examples

```
isochrone_png(x = 896488, y = 6006502, key = "abcd1234") %>%
    save_png(tempfile(pattern = "file", fileext = ".png"))
```

Index

esri_to_sgdf, 2, *4, 5*

get_stops, 2, *10*

isochrone, 3
isochrone_esri, *2*, 5, *8*
isochrone_png, 6, *11*
isochrone_pois, 7, *9*

pixel_walktimes, *4, 5*, 8
pois_walktimes, *4*, *7*, 9
pubtrans_ch_nearby, *2*, 10

response, *4–7*, *10*

save_png, *4*, *6*, 11
SpatialGridDataFrame-class, *2*, *5*