Package 'opensensmapr'

June 15, 2018

Type Package

Title Client for the Data API of openSenseMap.org

Version 0.4.1

URL http://github.com/noerw/opensensmapR

BugReports http://github.com/noerw/opensensmapR/issues

Imports dplyr, httr, digest, magrittr

Suggests maps, maptools, readr, tibble, rgeos, sf, knitr, rmarkdown, lubridate, units, jsonlite, ggplot2, zoo, lintr, testthat, covr

Description Download environmental measurements and sensor station metadata from the API of open data sensor web platform <https://opensensemap.org> for analysis in R.
This platform provides real time data of more than 1500 low-cost sensor stations for PM10, PM2.5, temperature, humidity, UV-A intensity and more phenomena.
The package aims to be compatible with 'sf' and the 'Tidyverse', and provides several helper functions for data exploration and transformation.

License GPL (>= 2) | file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Norwin Roosen [aut, cre],
Daniel Nuest [ctb] (<https://orcid.org/0000-0003-2392-6140>)

Maintainer Norwin Roosen <hello@nroo.de>

Repository CRAN

Date/Publication 2018-06-15 14:38:22 UTC

R topics documented:

`filter.osem_measurements` *Return rows with matching conditions, while maintaining class & attributes*

Description

Return rows with matching conditions, while maintaining class & attributes

Usage

```
filter.osem_measurements(.data, ..., .dots)
```

Arguments

`.data`	A osem_measurements data.frame to filter
`...`	other arguments
`.dots`	see corresponding function in package `dplyr`

See Also

`filter`

`filter.sensebox` *Return rows with matching conditions, while maintaining class & attributes*

Description

Return rows with matching conditions, while maintaining class & attributes

Usage

```
filter.sensebox(.data, ..., .dots)
```

Arguments

`.data`	A sensebox data.frame to filter
`...`	other arguments
`.dots`	see corresponding function in package `dplyr`

See Also

`filter`

`mutate.osem_measurements` *Add new variables to the data, while maintaining class & attributes*

Description

Add new variables to the data, while maintaining class & attributes

Usage

```
mutate.osem_measurements(.data, ..., .dots)
```

Arguments

`.data`	A osem_measurements data.frame to mutate
`...`	other arguments
`.dots`	see corresponding function in package `dplyr`

See Also

`mutate`

`mutate.sensebox` *Add new variables to the data, while maintaining class & attributes*

Description

Add new variables to the data, while maintaining class & attributes

Usage

```
mutate.sensebox(.data, ..., .dots)
```

Arguments

`.data`	A sensebox data.frame to mutate
`...`	other arguments
`.dots`	see corresponding function in package `dplyr`

See Also

`mutate`

`opensensmapr` *opensensmapr: Get sensor data from opensensemap.org*

Description

The opensensmapr package provides functions for

- retrieval of senseBox metadata,
- retrieval of senseBox measurements,
- general statistics about the openSenseMap database.

Additionally, helper functions are provided to ease the integration with the `sf` package for spatial analysis as well as `dplyr` for general data handling.

Retrieving senseBox metadata

On the openSenseMap, measurements are provided by sensors which are assigned to a sensor station ("senseBox"). A senseBox consists of a collection of sensors, a location (-history), an ID, as well as metadata about its owner & placement. senseBoxes can be retrieved either by ID, or as a collection with optional filters on their metadata

- `osem_box`: Get metadata about a single box by its ID.
- `osem_boxes`: Get metadata about all boxes, optionally filtered by their attributes.

The data is returned as a data.frame with the class sensebox attached. To help in getting an overview of the dataset additional functions are implemented:

- summary.sensebox(): Aggregate the metadata about the given list of senseBoxes.
- plot.sensebox(): Shows the spatial distribution of the given list of senseBoxes on a map. Requires additional packages!
- osem_phenomena: Get a named list with counts of the measured phenomena of the given list of senseBoxes.

Retrieving measurements

Measurements can be retrieved through osem_measurements for a given phenomenon only. A subset of measurements may be selected by

- a list of senseBoxes, previously retrieved through osem_box or osem_boxes.
- a geographic bounding box, which can be generated with the sf package.
- a time frame
- a exposure type of the given box

Data is returned as tibble with the class osem_measurements.

Retrieving statistics

Count statistics about the database are provided with osem_counts.

Integration with other packages

The package aims to be compatible with the tidyverse. Helpers are implemented to ease the further usage of the retrieved data:

- osem_as_sensebox & osem_as_measurements: Transform a foreign object to a sensebox data.frame or osem_measurements by attaching the required classes and attributes.
- st_as_sf.sensebox & st_as_sf.osem_measurements: Transform the senseBoxes or measurements into an sf compatible format for spatial analysis.
- filter.sensebox() & mutate.sensebox(): for use with dplyr.

Author(s)

Maintainer: Norwin Roosen <hello@nroo.de>

Other contributors:

- Daniel Nuest <daniel.nuest@uni-muenster.de> (0000-0003-2392-6140) [contributor]

See Also

Report bugs at https://github.com/noerw/opensensmapR/issues

openSenseMap API: https://api.opensensemap.org/

official openSenseMap API documentation: https://docs.opensensemap.org/

osem_as_measurements *Converts a foreign object to an osem_measurements data.frame.*

Description

Converts a foreign object to an osem_measurements data.frame.

Usage

```
osem_as_measurements(x)
```

Arguments

x	A data.frame to attach the class to

osem_as_sensebox *Converts a foreign object to a sensebox data.frame.*

Description

Converts a foreign object to a sensebox data.frame.

Usage

```
osem_as_sensebox(x)
```

Arguments

x	A data.frame to attach the class to

osem_box *Get a single senseBox by its ID*

Description

Get a single senseBox by its ID

Usage

```
osem_box(boxId, endpoint = osem_endpoint(), cache = NA)
```

Arguments

boxId	A string containing a senseBox ID
endpoint	The URL of the openSenseMap API instance
cache	Whether to cache the result, defaults to false. If a valid path to a directory is given, the response will be cached there. Subsequent identical requests will return the cached data instead.

Value

A `sensebox data.frame` containing a box in each row

See Also

openSenseMap API documentation (web)

`osem_phenomena`

`osem_boxes`

`osem_clear_cache`

Examples

```
# get a specific box by ID
b = osem_box('57000b8745fd40c8196ad04c')

# get a specific box by ID from a custom (selfhosted) openSenseMap API
b = osem_box('51030b8725fd30c2196277da', 'http://api.my-custom-osem.com')

# get a specific box by ID and cache the response, in order to provide
# reproducible results in the future.
b = osem_box('51030b8725fd30c2196277da', cache = tempdir())
```

---

osem_boxes *Get a set of senseBoxes from the openSenseMap*

Description

Boxes can be selected by a set of filters. Note that some filters do not work together:

1. `phenomenon` can only be applied together with `date` or `from / to`
2. `date` and `from / to` cannot be specified together

Usage

```
osem_boxes(exposure = NA, model = NA, grouptag = NA, date = NA,
  from = NA, to = NA, phenomenon = NA, endpoint = osem_endpoint(),
  progress = TRUE, cache = NA)
```

Arguments

exposure	Only return boxes with the given exposure ('indoor', 'outdoor', 'mobile')
model	Only return boxes with the given model
grouptag	Only return boxes with the given grouptag
date	Only return boxes that were measuring within ±4 hours of the given time
from	Only return boxes that were measuring later than this time
to	Only return boxes that were measuring earlier than this time
phenomenon	Only return boxes that measured the given phenomenon in the time interval as specified through `date` or `from / to`
endpoint	The URL of the openSenseMap API instance
progress	Whether to print download progress information, defaults to `TRUE`
cache	Whether to cache the result, defaults to false. If a valid path to a directory is given, the response will be cached there. Subsequent identical requests will return the cached data instead.

Value

A `sensebox data.frame` containing a box in each row

See Also

openSenseMap API documentation (web)

`osem_phenomena`

`osem_box`

`osem_clear_cache`

Examples

```
# get *all* boxes available on the API
b = osem_boxes()

# get all boxes with grouptag 'ifgi' that are placed outdoors
b = osem_boxes(grouptag = 'ifgi', exposure = 'outdoor')

# get all boxes with model 'luftdaten_sds011_dht22'
b = osem_boxes(grouptag = 'ifgi')

# get all boxes that have measured PM2.5 in the last 4 hours
b = osem_boxes(date = Sys.time(), phenomenon = 'PM2.5')

# get all boxes that have measured PM2.5 between Jan & Feb 2018
library(lubridate)
b = osem_boxes(
  from = date('2018-01-01'),
```

```
  to = date('2018-02-01'),
  phenomenon = 'PM2.5'
)

# get all boxes from a custom (selfhosted) openSenseMap API
b = osem_box(endpoint = 'http://api.my-custom-osem.com')

# get all boxes and cache the response, in order to provide
# reproducible results in the future. Also useful for development
# to avoid repeated loading times!
b = osem_boxes(cache = getwd())
b = osem_boxes(cache = getwd())

# get *all* boxes available on the API, without showing download progress
b = osem_boxes(progress = FALSE)
```

`osem_clear_cache` *Purge cached responses from the given cache directory*

Description

Purge cached responses from the given cache directory

Usage

```
osem_clear_cache(location = tempdir())
```

Arguments

`location`	A path to the cache directory, defaults to the sessions' `tempdir()`

Value

Boolean whether the deletion was successful

Examples

```
osem_boxes(cache = tempdir())
osem_clear_cache()

cachedir = paste(getwd(), 'osemcache', sep = '/')
osem_boxes(cache = cachedir)
osem_clear_cache(cachedir)
```

osem_counts *Get count statistics of the openSenseMap Instance*

Description

Provides information on number of senseBoxes, measurements, and measurements per minute.

Usage

```
osem_counts(endpoint = osem_endpoint(), cache = NA)
```

Arguments

endpoint	The URL of the openSenseMap API
cache	Whether to cache the result, defaults to false. If a valid path to a directory is given, the response will be cached there. Subsequent identical requests will return the cached data instead.

Details

Note that the API caches these values for 5 minutes.

Value

A named `list` containing the counts

See Also

openSenseMap API documentation (web)

osem_endpoint *Get the default openSenseMap API endpoint*

Description

Get the default openSenseMap API endpoint

Usage

```
osem_endpoint()
```

Value

A character string with the HTTP URL of the openSenseMap API

`osem_measurements` *Get the Measurements of a Phenomenon on opensensemap.org*

Description

Measurements can be retrieved either for a set of boxes, or through a spatial bounding box filter. To get all measurements, the `default` function applies a bounding box spanning the whole world.

Usage

```
osem_measurements(x, ...)

## Default S3 method:
osem_measurements(x, ...)

## S3 method for class 'bbox'
osem_measurements(x, phenomenon, exposure = NA, from = NA,
  to = NA, columns = NA, ..., endpoint = osem_endpoint(), progress = T,
  cache = NA)

## S3 method for class 'sensebox'
osem_measurements(x, phenomenon, exposure = NA,
  from = NA, to = NA, columns = NA, ..., endpoint = osem_endpoint(),
  progress = T, cache = NA)
```

Arguments

`x`	Depending on the method, either 1. a `chr` specifying the phenomenon, see `phenomenon` 2. a `st_bbox` to select sensors spatially, 3. a `sensebox` `data.frame` to select boxes from which measurements will be retrieved,
`...`	see parameters below
`phenomenon`	The phenomenon to retrieve measurements for
`exposure`	Filter sensors by their exposure ('indoor', 'outdoor', 'mobile')
`from`	A `POSIXt` like object to select a time interval
`to`	A `POSIXt` like object to select a time interval
`columns`	Select specific column in the output (see openSenseMap API documentation)
`endpoint`	The URL of the openSenseMap API
`progress`	Whether to print download progress information
`cache`	Whether to cache the result, defaults to false. If a valid path to a directory is given, the response will be cached there. Subsequent identical requests will return the cached data instead.

Value

An `osem_measurements data.frame` containing the requested measurements

Methods (by class)

- `default`: Get measurements from **all** senseBoxes.
- `bbox`: Get measurements by a spatial filter.
- `sensebox`: Get measurements from a set of senseBoxes.

See Also

openSenseMap API documentation (web)

`osem_box`

`osem_boxes`

`osem_clear_cache`

Examples

```
# get measurements from all boxes on the phenomenon 'PM10' from the last 48h
m = osem_measurements('PM10')

# get measurements from all mobile boxes on the phenomenon 'PM10' from the last 48h
m = osem_measurements('PM10', exposure = 'mobile')

# get measurements and cache them locally in the working directory.
# subsequent identical requests will load from the cache instead, ensuring
# reproducibility and saving time and bandwidth!
m = osem_measurements('PM10', exposure = 'mobile', cache = getwd())
m = osem_measurements('PM10', exposure = 'mobile', cache = getwd())

# get measurements returning a custom selection of columns
m = osem_measurements('PM10', exposure = 'mobile', columns = c(
  'value',
  'boxId',
  'sensorType',
  'lat',
  'lon',
  'height'
))


# get measurements from sensors within a custom WGS84 bounding box
bbox = structure(c(7, 51, 8, 52), class = 'bbox')
m = osem_measurements(bbox, 'Temperatur')

# construct a bounding box 12km around berlin using the sf package,
# and get measurements from stations within that box
library(sf)
bbox2 = st_point(c(13.4034, 52.5120)) %>%
```

```
  st_sfc(crs = 4326) %>%
  st_transform(3857) %>% # allow setting a buffer in meters
  st_buffer(set_units(12, km)) %>%
  st_transform(4326) %>% # the opensensemap expects WGS 84
  st_bbox()
m = osem_measurements(bbox2, 'Temperatur', exposure = 'outdoor')

# construct a bounding box from two points,
# and get measurements from stations within that box
points = st_multipoint(matrix(c(7.5, 7.8, 51.7, 52), 2, 2))
bbox3 = st_bbox(points)
m = osem_measurements(bbox2, 'Temperatur', exposure = 'outdoor')


# get measurements from a set of boxes
b = osem_boxes(grouptag = 'ifgi')
m4 = osem_measurements(b, phenomenon = 'Temperatur')

# ...or a single box
b = osem_box('57000b8745fd40c8196ad04c')
m5 = osem_measurements(b, phenomenon = 'Temperatur')

# get measurements from a single box on the from the last 40 days.
# requests are paged for long time frames, so the APIs limitation
# does not apply!
library(lubridate)
m1 = osem_measurements(
  b,
  'Temperatur',
  to = now(),
  from = now() - days(40)
)
```

---

osem_phenomena *Get the counts of sensors for each observed phenomenon.*

Description

Get the counts of sensors for each observed phenomenon.

Usage

```
osem_phenomena(boxes)

## S3 method for class 'sensebox'
osem_phenomena(boxes)
```

Arguments

boxes	A `sensebox` `data.frame` of boxes

Value

A named `list` containing the count of sensors observing a phenomenon per phenomenon

Methods (by class)

- `sensebox`: Get counts of sensors observing each phenomenon from a set of senseBoxes.

See Also

`osem_boxes`

Examples

```
# get the phenomena for a single senseBox
osem_phenomena(osem_box('593bcd656ccf3b0011791f5a'))

# get the phenomena for a group of senseBoxes
osem_phenomena(
  osem_boxes(grouptag = 'ifgi', exposure = 'outdoor', date = Sys.time())
)

# get phenomena with at least 30 sensors on opensensemap

  phenoms = osem_phenomena(osem_boxes())
  names(phenoms[phenoms > 29])
```

---

st_as_sf.osem_measurements

Convert a `osem_measurements` *dataframe to an* `st_sf` *object.*

Description

Convert a `osem_measurements` dataframe to an `st_sf` object.

Usage

```
st_as_sf.osem_measurements(x, ...)
```

Arguments

x	The object to convert
...	maybe more objects to convert

Value

The object with an st_geometry column attached.

st_as_sf.sensebox *Convert a* sensebox *dataframe to an* st_sf *object.*

Description

Convert a sensebox dataframe to an st_sf object.

Usage

```
st_as_sf.sensebox(x, ...)
```

Arguments

x	The object to convert
...	maybe more objects to convert

Value

The object with an st_geometry column attached.

Index