

Package ‘opensensmapr’

March 9, 2023

Type Package

Title Client for the Data API of 'openSenseMap.org'

Version 0.6.0

URL <https://github.com/sensebox/opensensmapR>

BugReports <https://github.com/sensebox/opensensmapR/issues>

Depends R (>= 3.5.0)

Imports dplyr, httr, digest, lazyeval, readr, purrr, magrittr

Suggests maps, maptools, 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)

Encoding UTF-8

RoxygenNote 7.2.3

VignetteBuilder knitr

NeedsCompilation no

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

Maintainer Jan Stenkamp <jan.stenkamp@uni-muenster.de>

Repository CRAN

Date/Publication 2023-03-09 09:00:07 UTC

R topics documented:

archive_fetch_measurements	2
filter.osem_measurements	3
filter.sensebox	3
mutate.osem_measurements	4
mutate.sensebox	4
opensensmapr	5
osem_archive_endpoint	7
osem_as_measurements	7
osem_as_sensebox	8
osem_box	8
osem_boxes	9
osem_box_to_archivename	11
osem_clear_cache	12
osem_counts	12
osem_endpoint	13
osem_ensure_api_available	13
osem_ensure_archive_available	14
osem_measurements	14
osem_measurements_archive	17
osem_phenomena	19
st_as_sf.osem_measurements	20
st_as_sf.sensebox	20
Index	21

archive_fetch_measurements

fetch measurements from archive from a single box, and a single sensor

Description

fetch measurements from archive from a single box, and a single sensor

Usage

```
archive_fetch_measurements(box, sensorId, fromDate, toDate, progress)
```

Arguments

box	A sensebox data.frame with a single box
sensorId	Character specifying the sensor
fromDate	Start date for measurement download, must be convertible via ‘as.Date’.
toDate	End date for measurement download (inclusive).
progress	whether to print progress

Value

A `tbl_df` containing observations of all selected sensors for each time stamp.

```
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

<code>.data</code>	A <code>osem_measurements</code> data.frame to filter
<code>...</code>	other arguments
<code>.dots</code>	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

<code>.data</code>	A <code>sensebox</code> data.frame to filter
<code>...</code>	other arguments
<code>.dots</code>	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

<code>.data</code>	A <code>osem_measurements</code> data.frame to mutate
<code>...</code>	other arguments
<code>.dots</code>	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

<code>.data</code>	A <code>sensebox</code> data.frame to mutate
<code>...</code>	other arguments
<code>.dots</code>	see corresponding function in package dplyr

See Also

[mutate](#)

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

There are two ways to retrieve measurements:

- [osem_measurements_archive](#): Downloads measurements for a *single box* from the openSenseMap archive. This function does not provide realtime data, but is suitable for long time frames.
- [osem_measurements](#): This function retrieves (realtime) measurements from the API. It works for a *single phenomenon* only, but provides various filters to select sensors 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

Use this function with caution for long time frames, as the API becomes quite slow is limited to 10.000 measurements per 30 day interval.

Data is returned as `tibble` with the class `osem_measurements`.

Retrieving statistics

Count statistics about the database are provided with `osem_counts`.

Using a different API instance / endpoint

You can override the functions `osem_endpoint` and `osem_endpoint_archive` inside the package namespace:

```
assignInNamespace("osem_endpoint", function() "http://mynewosem.org", "opensensmapr")
```

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: Jan Stenkamp <jan.stenkamp@uni-muenster.de> [contributor]

Authors:

- Norwin Roosen <hello@nroo.de>

Other contributors:

- Daniel Nuest <daniel.nuest@uni-muenster.de> ([ORCID](#)) [contributor]

See Also

Report bugs at <https://github.com/sensebox/opensensmapR/issues>

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

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

osem_archive_endpoint *Returns the default endpoint for the archive *download* While the front end domain is archive.opensensemap.org, file downloads are provided via sciebo.*

Description

Returns the default endpoint for the archive *download* While the front end domain is archive.opensensemap.org, file downloads are provided via sciebo.

Usage

```
osem_archive_endpoint()
```

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. Should have at least a 'value' and 'createdAt' column.

Value

data.frame of class osem_measurements

osem_as_sensebox	<i>Converts a foreign object to a sensebox data.frame.</i>
------------------	--

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
---	-------------------------------------

Value

data.frame of class sensebox

osem_box	<i>Get a single senseBox by its ID</i>
----------	--

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

```
## Not run:
# 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())

## End(Not run)
```

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,
  bbox = 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
bbox	Only return boxes that are within the given boundingbox, vector of 4 WGS84 coordinates. Order is: longitude southwest, latitude southwest, longitude northeast, latitude northeast. Minimal and maximal values are: -180, 180 for longitude and -90, 90 for latitude.
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

```
## Not run:
# 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)

## End(Not run)

```

```
osem_box_to_archivename
```

replace chars in box name according to archive script: <https://github.com/sensebox/osem-archiver/blob/612e14b/helpers.sh#L66>

Description

replace chars in box name according to archive script: <https://github.com/sensebox/osem-archiver/blob/612e14b/helpers.sh#L66>

Usage

```
osem_box_to_archivename(box)
```

Arguments

box A sensebox data.frame

Value

character with archive identifier for each box

osem_clear_cache	<i>Purge cached responses from the given cache directory</i>
------------------	--

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

```
## Not run:
osem_boxes(cache = tempdir())
osem_clear_cache()

cachedir = paste(getwd(), 'osemcache', sep = '/')
dir.create(file.path(cachedir), showWarnings = FALSE)
osem_boxes(cache = cachedir)
osem_clear_cache(cachedir)

## End(Not run)
```

osem_counts	<i>Get count statistics of the openSenseMap Instance</i>
-------------	--

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	<i>Get the default openSenseMap API endpoint</i>
---------------	--

Description

Get the default openSenseMap API endpoint

Usage

```
osem_endpoint()
```

Value

A character string with the HTTP URL of the openSenseMap API

osem_ensure_api_available	<i>Check if the given openSenseMap API endpoint is available</i>
---------------------------	--

Description

Check if the given openSenseMap API endpoint is available

Usage

```
osem_ensure_api_available(endpoint = osem_endpoint())
```

Arguments

endpoint The API base URL to check, defaulting to [osem_endpoint](#)

Value

TRUE if the API is available, otherwise `stop()` is called.

osem_ensure_archive_available

Check if the given openSenseMap archive endpoint is available

Description

Check if the given openSenseMap archive endpoint is available

Usage

```
osem_ensure_archive_available(endpoint = osem_archive_endpoint())
```

Arguments

endpoint The archive base URL to check, defaulting to [osem_archive_endpoint](#)

Value

TRUE if the archive is available, otherwise `stop()` is called.

osem_measurements

Fetch 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 = TRUE,
    cache = NA
)

## S3 method for class 'sensebox'
osem_measurements(
  x,
  phenomenon,
  exposure = NA,
  from = NA,
  to = NA,
  columns = NA,
  ...,
  endpoint = osem_endpoint(),
  progress = TRUE,
  cache = NA
)

```

Arguments

x	Depending on the method, either <ol style="list-style-type: none"> 1. a chr specifying the phenomenon, see phenomenon 2. a <code>st_bbox</code> to select sensors spatially, 3. a <code>sensebox</code> 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 <code>openSenseMap</code> API documentation)
endpoint	The URL of the <code>openSenseMap</code> 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)

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

See Also

[openSenseMap API documentation \(web\)](#)

[osem_box](#)

[osem_boxes](#)

[osem_clear_cache](#)

Examples

```
## Not run:
# 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'
))

## End(Not run)
## Not run:
# 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)
library(units)
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')

## End(Not run)

# 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('5700b8745fd40c8196ad04c')
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_measurements_archive
```

Fetch day-wise measurements for a single box from the openSenseMap archive.

Description

This function is significantly faster than `osem_measurements` for large time-frames, as daily CSV dumps for each sensor from archive.opensensemap.org are used. Note that the latest data available is from the previous day.

Usage

```

osem_measurements_archive(x, ...)

## S3 method for class 'sensebox'
osem_measurements_archive(

```

```

x,
fromDate,
toDate = fromDate,
sensorFilter = ~TRUE,
...,
progress = TRUE
)

```

Arguments

x	A 'sensebox data.frame' of a single box, as retrieved via osem_box , to download measurements for.
...	see parameters below
fromDate	Start date for measurement download, must be convertible via 'as.Date'.
toDate	End date for measurement download (inclusive).
sensorFilter	A NSE formula matching to x\$sensors, selecting a subset of sensors.
progress	Whether to print download progress information, defaults to TRUE.

Details

By default, data for all sensors of a box is fetched, but you can select a subset with a [dplyr](#)-style NSE filter expression.

The function will warn when no data is available in the selected period, but continue the remaining download.

Value

A `tbl_df` containing observations of all selected sensors for each time stamp.

Methods (by class)

- `osem_measurements_archive(sensebox)`: Get daywise measurements for one or more sensors of a single box.

See Also

[openSenseMap archive](#)

[osem_measurements](#)

[osem_box](#)

Examples

```

# fetch measurements for a single day
box = osem_box('593bcd656ccf3b0011791f5a')
m = osem_measurements_archive(box, as.POSIXlt('2018-09-13'))

# fetch measurements for a date range and selected sensors

```

```
sensors = ~ phenomenon %in% c('Temperatur', 'Beleuchtungsstärke')
m = osem_measurements_archive(
  box,
  as.POSIXlt('2018-09-01'), as.POSIXlt('2018-09-30'),
  sensorFilter = sensors
)
```

osem_phenomena	<i>Get the counts of sensors for each observed phenomenon.</i>
----------------	--

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)

- `osem_phenomena(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

archive_fetch_measurements, 2

data.frame, 5
dplyr, 3–6, 18

filter, 3
filter.osem_measurements, 3
filter.sensebox, 3

mutate, 4
mutate.osem_measurements, 4
mutate.sensebox, 4

opensensmapr, 5
opensensmapr-package (opensensmapr), 5
osem_archive_endpoint, 7, 14
osem_as_measurements, 6, 7
osem_as_sensebox, 6, 8
osem_box, 5, 8, 10, 16, 18
osem_box_to_archivename, 11
osem_boxes, 5, 9, 9, 16, 19
osem_clear_cache, 9, 10, 12, 16
osem_counts, 6, 12
osem_endpoint, 13, 14
osem_ensure_api_available, 13
osem_ensure_archive_available, 14
osem_measurements, 5, 14, 17, 18
osem_measurements_archive, 5, 17
osem_phenomena, 5, 9, 10, 19

sf, 5, 6
st_as_sf.osem_measurements, 6, 20
st_as_sf.sensebox, 6, 20
st_bbox, 15
st_sf, 20