

Package ‘PurpleAir’

September 22, 2025

Title Query the 'PurpleAir' Application Programming Interface

Version 1.1.0

Description Send requests to the 'PurpleAir' Application Programming Interface (API; <<https://community.purpleair.com/c/data/api/18>>). Check a 'PurpleAir' API key and get information about the related organization. Download real-time data from a single 'PurpleAir' sensor or many sensors by sensor identifier, geographical bounding box, or time since modified. Download historical data from a single sensor. Stream real time data from monitors on a local area network.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Suggests sf, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/cole-brokamp/PurpleAir>

BugReports <https://github.com/cole-brokamp/PurpleAir/issues>

Imports httr2, purrr (>= 1.1.0), tibble, rlang, cli, dplyr, glue

Depends R (>= 4.1.0)

NeedsCompilation no

Author Cole Brokamp [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-0289-3151>>)

Maintainer Cole Brokamp <cole@colebrokamp.com>

Repository CRAN

Date/Publication 2025-09-22 14:30:11 UTC

Contents

check_api_key	2
find_local_pam	3
get_organization_data	3
get_sensors_data	4

get_sensor_data	5
get_sensor_history	6
ip_pam_id	7
local_sensor_data	8
local_sensor_live	9
Index	10

check_api_key	<i>Check Purple Air API Key</i>
---------------	---------------------------------

Description

Use the PurpleAir API to validate your Purple Air API Key. Find more details on this function at <https://api.purpleair.com/#api-keys-check-api-key>. Storing your key in the environment variable PURPLE_AIR_API_KEY is safer than storing it in source code and is used by default in each PurpleAir function.

Usage

```
check_api_key(purple_air_api_key = Sys.getenv("PURPLE_AIR_API_KEY"))
```

Arguments

purple_air_api_key
A character that is your PurpleAir API READ key

Value

If the key is valid, a message is emitted and the input is invisibly returned; invalid keys will throw an R error which utilizes information from the underlying http error to inform the user.

See Also

get_organization_data

Examples

```
## Not run:  
check_api_key()  
try(check_api_key("foofy"))  
  
## End(Not run)
```

find_local_pam	<i>find purple area monitors on a local network</i>
----------------	---

Description

All IP addresses within the network are pinged to possibly return a purple air monitor sensor ID.

Usage

```
find_local_pam(network_prefix = "192.168.1", timeout = 1)
```

Arguments

network_prefix	character string; base IPv4 prefix (first three octets) used to generate IP addresses
timeout	numeric; number of seconds to wait for each ping

Details

If the mirai package is available, this function will ensure that at least version 1.1.0 of the purrr package is installed to scan the network in parallel, according to `mirai::daemons()` set by the user. This reduces the time it takes, but does not use a progress bar.

Value

a list of purple air monitor IP addresses named according to their Sensor IDs

Examples

```
## Not run:
mirai::daemons(12)
find_local_pam()
mirai::daemons(0)

## End(Not run)
```

get_organization_data	<i>Get Organization Data</i>
-----------------------	------------------------------

Description

Use the PurpleAir API to retrieve information for the organization containing the provided `api_key`. Find more details on this function at <https://api.purpleair.com/#api-organization-get-organization-data>

Usage

```
get_organization_data(purple_air_api_key = Sys.getenv("PURPLE_AIR_API_KEY"))
```

Arguments

purple_air_api_key
A character that is your PurpleAir API READ key

Value

A list of organization info

See Also

check_api_key

Examples

```
## Not run:
get_organization_data()

## End(Not run)
```

get_sensors_data	<i>Get Sensors Data</i>
------------------	-------------------------

Description

Retrieves the latest data of multiple sensors matching the provided parameters. Find more details on sensor fields at <https://api.purpleair.com/#api-sensors-get-sensors-data>.

Usage

```
get_sensors_data(
  x,
  fields,
  location_type = c("both", "inside", "outside"),
  max_age = as.integer(604800),
  purple_air_api_key = Sys.getenv("PURPLE_AIR_API_KEY"),
  read_keys = NULL
)
```

Arguments

x an input object used to define multiple sensors:

- an integer (or numeric or character) vector will select sensors based on sensor_index (API: show_only)
- a st_bbox object will select sensors geographically (API: nwlat, nwlon, selat, selon)
- a POSIXct object will select sensors modified since the given time (API: modified_since)

fields	A character vector of which 'sensor data fields' to return
location_type	character; restrict to only "outside" or "inside" sensors (Outside: 0, Inside: 1)
max_age	integer; filter results to only include sensors modified or updated within the last number of seconds
purple_air_api_key	Your PurpleAir API READ key
read_keys	A character vector of keys required to read data from private devices

Value

A list of sensor data, named by the provided fields

See Also

get_sensor_data

Examples

```
## Not run:
# get sensors data by integer, numeric, or character vector of `sensor_index`
get_sensors_data(
  x = as.integer(c(175257, 175413)),
  fields = c("name", "last_seen", "pm2.5_cf_1", "pm2.5_atm")
)
get_sensors_data(
  x = c(175257, 175413),
  fields = c("name", "last_seen", "pm2.5_cf_1", "pm2.5_atm")
)
get_sensors_data(
  x = c("175257", "175413"),
  fields = c("name"), location_type = "outside"
)
# get sensors by bounding box around Hamilton County, OH
sf::st_bbox(c("xmin" = -84.82030, "ymin" = 39.02153,
              "xmax" = -84.25633, "ymax" = 39.31206),
  crs = 4326) |>
  get_sensors_data(fields = c("name"))
# sensors modified in the last 60 seconds
get_sensors_data(as.POSIXct(Sys.time()) - 60, fields = "name")

## End(Not run)
```

get_sensor_data

Get Sensor Data

Description

Retrieves the latest data of a single sensor matching the provided sensor_index. Find more details on sensor fields at <https://api.purpleair.com/#api-sensors-get-sensor-data>.

Usage

```
get_sensor_data(
  sensor_index,
  fields,
  purple_air_api_key = Sys.getenv("PURPLE_AIR_API_KEY"),
  read_key = NULL
)
```

Arguments

<code>sensor_index</code>	Integer (or numeric, character object coerceable to integer) <code>sensor_index</code>
<code>fields</code>	A character vector of which 'sensor data fields' to return
<code>purple_air_api_key</code>	A character that is your PurpleAir API READ key
<code>read_key</code>	A character key required to read data from private devices

Value

A list of sensor data, named by the provided fields

See Also

`get_sensors_data` `get_sensor_history`

Examples

```
## Not run:
get_sensor_data(sensor_index = 175413, fields = c("name", "last_seen", "pm2.5_cf_1", "pm2.5_atm"))
get_sensor_data(sensor_index = "175413", fields = c("name", "last_seen", "pm2.5_cf_1", "pm2.5_atm"))

## End(Not run)
```

<code>get_sensor_history</code>	<i>get sensor history</i>
---------------------------------	---------------------------

Description

Retrieves the latest history of a single sensor matching the provided `sensor_index`. Find more details on sensor fields at <https://api.purpleair.com/#api-sensors-get-sensor-history>.

Usage

```
get_sensor_history(
  sensor_index,
  fields,
  start_timestamp,
  end_timestamp,
```

```
average = c("10min", "30min", "60min", "6hr", "1day", "1week", "1month", "1year",
            "real-time"),
purple_air_api_key = Sys.getenv("PURPLE_AIR_API_KEY"),
read_key = NULL
)
```

Arguments

- sensor_index Integer (or numeric, character object coerceable to integer) sensor_index
- fields A character vector of which 'sensor data fields' to return
- start_timestamp time stamp of first required history entry (inclusive)
- end_timestamp end time stamp of history to return (exclusive)
- average time frame to request averaged results for
- purple_air_api_key A character that is your PurpleAir API READ key
- read_key A character key required to read data from private devices

Value

a list of sensor data, named by the provided fields

Examples

```
## Not run:
get_sensor_history(
  sensor_index = 175413,
  fields = c("pm1.0_cf_1", "pm1.0_atm", "pm2.5_cf_1", "pm2.5_atm"),
  start_timestamp = as.POSIXct("2024-07-02"),
  end_timestamp = as.POSIXct("2024-07-05")
)

## End(Not run)
```

ip_pam_id	<i>get purple air monitor id from ip address</i>
-----------	--

Description

get purple air monitor id from ip address

Usage

```
ip_pam_id(ip_address, timeout = 1)
```

Arguments

<code>ip_address</code>	character; address to send sensor id request to
<code>timeout</code>	numeric; number of seconds to wait for each ping

Value

NULL if address doesn't respond to a SensorId request; if ip address is a purple air monitor, the monitor id is returned

Examples

```
ip_pam_id("192.168.1.144") # purple air
ip_pam_id("192.168.1.148") # no server
ip_pam_id("192.168.1.141") # non-purple air
```

<code>local_sensor_data</code>	<i>local_sensor_data</i>
--------------------------------	--------------------------

Description

Get latest data (updated every two minutes) from a sensor the local area network.

Usage

```
local_sensor_data(ip_address)
```

Arguments

<code>ip_address</code>	address of purple air monitor on local area network to request data from
-------------------------	--

Value

a list of data returned by the sensor

Examples

```
## Not run:
local_sensor_data("192.168.1.144") |>
  _[c("DateTime", "current_temp_f", "current_humidity", "pm2_5_cf_1", "p25aqic")]

## End(Not run)
```

local_sensor_live	<i>local_sensor_live</i>
-------------------	--------------------------

Description

Stream the latest data from a sensor on the local area network. Data is updated every second on the device; this function waits half a second after each call, which takes less than half a second, ensuring sub one second updating frequency.

Usage

```
local_sensor_live(ip_address)
```

Arguments

ip_address	address of purple air monitor on local area network to request data from
------------	--

Examples

```
## Not run:  
local_sensor_live("192.168.1.144")  
  
## End(Not run)
```

Index

check_api_key, [2](#)

find_local_pam, [3](#)

get_organization_data, [3](#)

get_sensor_data, [5](#)

get_sensor_history, [6](#)

get_sensors_data, [4](#)

ip_pam_id, [7](#)

local_sensor_data, [8](#)

local_sensor_live, [9](#)