

# Package ‘PurpleAir’

January 20, 2025

**Title** Query the 'PurpleAir' Application Programming Interface

**Version** 1.0.1

**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.

**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, tibble, rlang, cli, dplyr, glue

**NeedsCompilation** no

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

**Maintainer** Cole Brokamp <[cole@colebrokamp.com](mailto:cole@colebrokamp.com)>

**Repository** CRAN

**Date/Publication** 2024-10-08 18:40:02 UTC

## Contents

check_api_key . . . . .	2
get_organization_data . . . . .	2
get_sensors_data . . . . .	3
get_sensor_data . . . . .	5
get_sensor_history . . . . .	6

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

---

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)
```

---

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	<p>an input object used to define multiple sensors:</p> <ul style="list-style-type: none"> <li>• an integer (or numeric or character) vector will select sensors based on sensor_index (API: show_only)</li> <li>• a st_bbox object will select sensors geographically (API: nwlat, nwlon, selat, selon)</li> <li>• a POSIXct object will select sensors modified since the given time (API: modified_since)</li> </ul>
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	<i>Get Sensor Data</i>
-----------------	------------------------

---

### 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

sensor_index	Integer (or numeric, character object coerceable to integer) sensor_index
fields	A character vector of which 'sensor data fields' to return
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

### 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)
```

---

```
get_sensor_history    get sensor history
```

---

### 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

<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>start_timestamp</code>	time stamp of first required history entry (inclusive)
<code>end_timestamp</code>	end time stamp of history to return (exclusive)
<code>average</code>	time frame to request averaged results for
<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`

### 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)
```

# Index

`check_api_key`, [2](#)

`get_organization_data`, [2](#)

`get_sensor_data`, [5](#)

`get_sensor_history`, [6](#)

`get_sensors_data`, [3](#)