

# Package ‘ropenaq’

January 31, 2019

**Type** Package

**Title** Accesses Air Quality Data from the Open Data Platform OpenAQ

**Version** 0.2.7

**Description** Allows access to air quality data from the API of the OpenAQ platform <<https://docs.openaq.org/>>, with the different services the API offers (getting measurements for a given query, getting latest measurements, getting lists of available countries/cities/locations). This package has been peer-reviewed by rOpenSci (v. 0.1.0).

**License** GPL (>= 2)

**LazyData** TRUE

**RoxygenNote** 6.1.1

**URL** <http://github.com/ropensci/ropenaq>

**BugReports** <http://github.com/ropensci/ropenaq/issues>

**Suggests** knitr, rmarkdown, lintr, ggplot2, testthat, openair, devtools, rworldmap, viridis

**VignetteBuilder** knitr

**Imports** crul, lubridate, dplyr, jsonlite, tidyr, lazyeval

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Maëlle Salmon [aut, cre],  
Andrew MacDonald [rev] (Andrew MacDonald reviewed the package for rOpenSci, see <https://github.com/ropensci/onboarding/issues/24>),  
Andy Teucher [rev] (Andy Teucher reviewed the package for rOpenSci, see <https://github.com/ropensci/onboarding/issues/24>),  
Mikayla Murphy [ctb]

**Maintainer** Maëlle Salmon <[maelle.salmon@yahoo.se](mailto:maelle.salmon@yahoo.se)>

**Repository** CRAN

**Date/Publication** 2019-01-31 15:20:03 UTC

## R topics documented:

aq_cities . . . . .	2
aq_countries . . . . .	3
aq_latest . . . . .	4
aq_locations . . . . .	6
aq_measurements . . . . .	8

<b>Index</b>	<b>11</b>
--------------	-----------

---

aq_cities	<i>Provides a simple listing of cities within the platform.</i>
-----------	---

---

### Description

Provides a simple listing of cities within the platform.

### Usage

```
aq_cities(country = NULL, limit = 10000, page = NULL)
```

### Arguments

country	Limit results by a certain country – a two-letters code see countries() for finding code based on name.
limit	Change the number of results returned, max is 10000.
page	The page of the results to query. This can be useful if e.g. there are 2000 measurements, then first use page=1 and page=2 with limit=100 to get all measurements for your query.

### Details

For queries involving a city argument, the URL-encoded name of the city (as in cityURL), not its name, should be used.

### Value

A results data.frame (dplyr "tbl\_df") with 5 columns:

- city name ("city"), country code ("country"),
- number of measures in total for the city ("count"),
- number of locations ("locations"),
- and also an URL encoded string for the city ("cityURL") which can be useful for queries involving a city argument.

and 2 attributes, a meta data.frame (dplyr "tbl\_df") with 1 line and 5 columns:

- the API name ("name"),

- the license of the data ("license"),
- the website url ("website"),
- the queried page ("page"),
- the limit on the number of results ("limit"),
- the number of results found on the platform for the query ("found")

and a timestamp data.frame (dplyr "tbl\_df") with the query time and the last time at which the data was modified on the platform.

### Examples

```
## Not run:
cities <- aq_cities(country = "BA")
cities

## End(Not run)
```

---

aq_countries	<i>Provides a simple listing of countries within the platform.</i>
--------------	--

---

### Description

Provides a simple listing of countries within the platform.

### Usage

```
aq_countries(limit = 10000, page = NULL)
```

### Arguments

limit	Change the number of results returned, max is 10000.
page	The page of the results to query. This can be useful if e.g. there are 2000 measurements, then first use page=1 and page=2 with limit=100 to get all measurements for your query.

### Details

For queries involving a country argument, the code of the country, not its name, should be used.

### Value

A results data.frame (dplyr "tbl\_df") with 3 columns:

- the number of measures for a country ("count"),
- the ISO 3166-1 alpha-2 code of the country ("code"),
- and its name ("name").

and 2 attributes, a meta data.frame (dplyr "tbl\_df") with 1 line and 5 columns:

- the API name ("name"),
- the license of the data ("license"),
- the website url ("website"),
- the queried page ("page"),
- the limit on the number of results ("limit"),
- the number of results found on the platform for the query ("found")

and a timestamp data.frame (dplyr "tbl\_df") with the query time and the last time at which the data was modified on the platform.

### Examples

```
## Not run:
countries <- aq_countries()
countries

## End(Not run)
```

---

aq_latest	<i>Provides the latest value of each available parameter for every location in the system.</i>
-----------	--

---

### Description

Provides the latest value of each available parameter for every location in the system.

### Usage

```
aq_latest(country = NULL, city = NULL, location = NULL,
  parameter = NULL, has_geo = NULL, limit = 10000, latitude = NULL,
  longitude = NULL, radius = NULL, page = NULL)
```

### Arguments

country	Limit results by a certain country – a two-letters code see countries() for finding code based on name.
city	Limit results by a certain city.
location	Limit results by a certain location.
parameter	Limit to only a certain parameter (valid values are 'pm25', 'pm10', 'so2', 'no2', 'o3', 'co' and 'bc'). If no parameter is given, all parameters are retrieved.
has_geo	has_geo Filter out items that have or do not have geographic information.
limit	Change the number of results returned, max is 10000.

latitude	Latitude of the center point (lat, lon) used to get measurements within a certain area.
longitude	Longitude of the center point (lat, lon) used to get measurements within a certain area.
radius	Radius (in meters) used to get measurements within a certain area, must be used with latitude and longitude
page	The page of the results to query. This can be useful if e.g. there are 2000 measurements, then first use page=1 and page=2 with limit=100 to get all measurements for your query.

### Details

For queries involving a city or location argument, the URL-encoded name of the city/location (as in cityURL/locationURL), not its name, should be used. You can query any nested combination of country/location/city (level 1, 2 and 3), with only one value for each argument. If you write inconsistent combination such as city="Paris" and country="IN", an error message will be returned. If you write city="Delhi", you do not need to write the code of the country, unless one day there is a city with the same name in another country.

### Value

A results data.frame (dplyr "tbl\_df") with 11 columns:

- the name of the location ("location"),
- the city it is in ("city"),
- the code of country it is in ("country"),
- its longitude ("longitude") and latitude if available ("latitude"),
- the parameter ("parameter")
- the value of the measurement ("value")
- the last time and date at which the value was updated ("lastUpdated"),
- the unit of the measure ("unit")
- and finally an URL encoded version of the city name ("cityURL")
- and of the location name ("locationURL").

. and two attributes, a meta data.frame (dplyr "tbl\_df") with 1 line and 5 columns:

- the API name ("name"),
- the license of the data ("license"),
- the website url ("website"),
- the queried page ("page"),
- the limit on the number of results ("limit"),
- the number of results found on the platform for the query ("found")

and a timestamp data.frame (dplyr "tbl\_df") with the query time and the last time at which the data was modified on the platform.

**Examples**

```
## Not run:
latest_chennai <- aq_latest(country='IN', city='Chennai')
latest_chennai
attr(latest_chennai, "meta")
attr(latest_chennai, "timestamp")
aq_latest(parameter='co')

## End(Not run)
```

---

aq\_locations

*Providing data about distinct measurement locations.*


---

**Description**

Providing data about distinct measurement locations.

**Usage**

```
aq_locations(country = NULL, city = NULL, location = NULL,
  parameter = NULL, has_geo = NULL, limit = 10000, latitude = NULL,
  longitude = NULL, radius = NULL, page = NULL)
```

**Arguments**

country	Limit results by a certain country – a two-letters code see countries() for finding code based on name.
city	Limit results by a certain city.
location	Limit results by a certain location.
parameter	Limit to only a certain parameter (valid values are 'pm25', 'pm10', 'so2', 'no2', 'o3', 'co' and 'bc'). If no parameter is given, all parameters are retrieved.
has_geo	has_geo Filter out items that have or do not have geographic information.
limit	Change the number of results returned, max is 10000.
latitude	Latitude of the center point (lat, lon) used to get measurements within a certain area.
longitude	Longitude of the center point (lat, lon) used to get measurements within a certain area.
radius	Radius (in meters) used to get measurements within a certain area, must be used with latitude and longitude
page	The page of the results to query. This can be useful if e.g. there are 2000 measurements, then first use page=1 and page=2 with limit=100 to get all measurements for your query.

## Details

For queries involving a city or location argument, the URL-encoded name of the city/location (as in cityURL/locationURL), not its name, should be used. You can query any nested combination of country/location/city (level 1, 2 and 3), with only one value for each argument. If you write inconsistent combination such as city="Paris" and country="IN", an error message will be returned. If you write city="Delhi", you do not need to write the code of the country, unless one day there is a city with the same name in another country.

## Value

A results data.frame (dplyr "tbl\_df") with 12 columns:

- the name of the location ("location"),
- the city it is in ("city"),
- the code of country it is in ("country"),
- the name of the source of the information ("sourceName"),
- the number of measures for this location in the platform ("count"),
- the last time and date at which measures were updated for this location ("lastUpdated"),
- the first time and date at which measures were updated for this location ("firstUpdated"),
- the parameters available for this location ("parameters"),
- its longitude ("longitude") and latitude if available ("latitude"),
- and finally an URL encoded version of the city name ("cityURL")
- and of the location name ("locationURL").

and two attributes, a meta data.frame (dplyr "tbl\_df") with 1 line and 5 columns:

- the API name ("name"),
- the license of the data ("license"),
- the website url ("website"),
- the queried page ("page"),
- the limit on the number of results ("limit"),
- the number of results found on the platform for the query ("found")

and a timestamp data.frame (dplyr "tbl\_df") with the query time and the last time at which the data was modified on the platform.

## Examples

```
## Not run:  
aq_locations(city='Delhi', parameter='co')  
  
## End(Not run)
```

---

aq_measurements	<i>Function for getting measurements table from the openAQ API</i>
-----------------	--

---

### Description

Function for getting measurements table from the openAQ API

### Usage

```
aq_measurements(country = NULL, city = NULL, location = NULL,
  parameter = NULL, has_geo = NULL, date_from = NULL,
  date_to = NULL, limit = 10000, value_from = NULL,
  latitude = NULL, longitude = NULL, radius = NULL,
  attribution = FALSE, averaging_period = FALSE, source_name = FALSE,
  value_to = NULL, page = NULL)
```

### Arguments

country	Limit results by a certain country – a two-letters code see countries() for finding code based on name.
city	Limit results by a certain city.
location	Limit results by a certain location.
parameter	Limit to only a certain parameter (valid values are 'pm25', 'pm10', 'so2', 'no2', 'o3', 'co' and 'bc'). If no parameter is given, all parameters are retrieved.
has_geo	has_geo Filter out items that have or do not have geographic information.
date_from	Show results after a certain date. (character year-month-day, ex. '2015-12-20'). Note, since November 2017 the API only provides access to the last three months so if you need more data you need to fetch it via Amazon S3 ( <a href="https://medium.com/@openaq/changes-to-the-openaq-api-and-how-to-access-the-full-archive-of-data-3324b136da8c">https://medium.com/@openaq/changes-to-the-openaq-api-and-how-to-access-the-full-archive-of-data-3324b136da8c</a> ), potentially using the aws.s3 package.
date_to	Show results before a certain date. (character year-month-day, ex. '2015-12-20')
limit	Change the number of results returned, max is 10000.
value_from	Show results above value threshold, useful in combination with parameter.
latitude	Latitude of the center point (lat, lon) used to get measurements within a certain area.
longitude	Longitude of the center point (lat, lon) used to get measurements within a certain area.
radius	Radius (in meters) used to get measurements within a certain area, must be used with latitude and longitude
attribution	Logical, whether to add a column with attribution information
averaging_period	Logical, whether to add a column with averaging_period information



source_name	Logical, whether to add a column with source_name information
value_to	Show results below value threshold, useful in combination with parameter.
page	The page of the results to query. This can be useful if e.g. there are 2000 measurements, then first use page=1 and page=2 with limit=100 to get all measurements for your query.

### Details

For queries involving a city or location argument, the URL-encoded name of the city/location (as in cityURL/locationURL), not its name, should be used. You can query any nested combination of country/location/city (level 1, 2 and 3), with only one value for each argument. If you write inconsistent combination such as city="Paris" and country="IN", an error message will be returned. If you write city="Delhi", you do not need to write the code of the country, unless one day there is a city with the same name in another country.

If you choose to get the attribution in the output, lines might be repeated as there might be several attributions.

### Value

A results data.frame (dplyr "tbl\_df") with at least 12 columns:

- the name of the location ("location"),
- the parameter ("parameter")
- the value of the measurement ("value")
- the unit of the measure ("unit")
- the code of country the location is in ("country"),
- the city it is in ("city"),
- and finally an URL encoded version of the city name ("cityURL")
- and of the location name ("locationURL"),
- the UTC POSIXct time ("dateUTC"),
- the local POSIXct time ("dateLocal"),
- its longitude ("longitude") and latitude if available ("latitude").

and two attributes, a meta data.frame (dplyr "tbl\_df") with 1 line and 5 columns:

- the API name ("name"),
- the license of the data ("license"),
- the website url ("website"),
- the queried page ("page"),
- the limit on the number of results ("limit"),
- the number of results found on the platform for the query ("found")

and a timestamp data.frame (dplyr "tbl\_df") with the query time and the last time at which the data was modified on the platform.

**Examples**

```
## Not run:
output <- aq_measurements(country='IN', limit=9, city='Chennai',
                           page = 1)

output
attr(output, "meta")
attr(output, "timestamp")

## End(Not run)
```

# Index

aq\_cities, 2  
aq\_countries, 3  
aq\_latest, 4  
aq\_locations, 6  
aq\_measurements, 8