Package: openaq (via r-universe)

May 20, 2026

Title Access Air Quality Data from the 'OpenAQ' API

Description Provides an interface to the 'OpenAQ' API <https://openaq.org/>, a platform for real-time and historical air quality data from around the world. Users can retrieve measurement data, metadata for sensors and locations for air quality research and monitoring.

Version 1.0.0

License MIT + file LICENSE

URL https://github.com/openaq/openaq-r, https://openaq.github.io/openaq-r/

BugReports https://github.com/openaq/openaq-r/issues

Depends R (>= 4.1.0)

Imports httr2 (>= 1.2.0), rstudioapi, methods, lubridate

Suggests testthat (>= 3.0.0), webmockr (>= 0.8.0), vcr (>= 2.1.0), withr (>= 3.0.2), knitr, maps, rmarkdown, sf (>= 0.9), spelling, jsonlite, mockery,

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Language en-US

Config/pak/sysreqs libssl-dev

Repository https://openaq.r-universe.dev

Date/Publication 2026-03-19 15:44:28 UTC

RemoteUrl https://github.com/openaq/openaq-r

RemoteRef HEAD

RemoteSha 5d83fb769d3fbf1f6857fe6d5fc01dff300c88e5

Contents

`as.data.frame.openaq_countries_list` *Method for converting openaq_countries_list to data frame.*

Description

Method for converting openaq_countries_list to data frame.

Usage

```
## S3 method for class 'openaq_countries_list'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A list of countries as returned from list_countries.
`row.names`	`NULL` or a character vector giving the row names for the data frame. Missing values are not allowed.
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
`...`	additional arguments to be passed to or from methods.

Value

A data frame class of the countries results, with the following columns:

id Numeric. The countries identifier

name Character. Then English name of the country.

code Character. The ISO-3166 Alpha 2 identifier for the country.

datetime_first POSIXct. The datetime of the first measurement value available in the country.

datetime_last POSIXct. The datetime of the last measurement value available in the country.

parameter_ids Character. A comma delimited list of parameter ids that are measured within the country.

parameter_names Character. a comma delimited list of parameter names and their units that are measured within the country.

The data frame also includes a `meta` attribute from the original `openaq_countries_list`.

Examples

```
countries <- list_countries(as_data_frame = FALSE)
as.data.frame(countries)
```

`as.data.frame.openaq_instruments_list` *Method for converting openaq_instruments_list to data frame.*

Description

Method for converting openaq_instruments_list to data frame.

Usage

```
## S3 method for class 'openaq_instruments_list'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A list of instruments as returned from list_instruments.
`row.names`	`NULL` or a character vector giving the row names for the data frame. Missing values are not allowed.
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
`...`	additional arguments to be passed to or from methods.

Value

A data frame class of the instruments results, with the following columns:

id Numeric. The instruments identifier.

name Character. The name of the measurement instrument.

is_monitor Logical. Indicates if the instrument is considered a reference monitor.

manufacturer_id Numeric. The manufacturers identifier for the manufacturer that makes the instrument.

manufacturer_name Factor. The name of manufacturer that makes the instrument.

The data frame also includes a `meta` attribute from the original `openaq_instruments_list`.

Examples

```
instruments <- list_instruments(as_data_frame = FALSE)
as.data.frame(instruments)
```

`as.data.frame.openaq_latest_list` *Method for converting openaq_latest_list to data frame.*

Description

Method for converting openaq_latest_list to data frame.

Usage

```
## S3 method for class 'openaq_latest_list'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A list of latest measurements as returned from list_location_latest or list_parameter_latest.
`row.names`	`NULL` or a character vector giving the row names for the data frame. Missing values are not allowed.
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
`...`	additional arguments to be passed to or from methods.

Value

A data frame class of the latest results, with the following columns:

sensors_id Numeric. The sensors identifier.

locations_id Numeric. The locations identifier.

value Numeric. The measurement value.

datetime_local POSIXct. The datetime of the measurement value, in local time

datetime_utc POSIXct. The datetime of the measurement value, in UTC time

latitude Numeric. The latitude, geographic Y, value for the measurement.

longitude Numeric. The longitude, geographic X, value for the measurement.

The data frame also includes a `meta` attribute from the original `openaq_latest_list`.

Examples

```
latest <- list_location_latest(2178, as_data_frame = FALSE)
as.data.frame(latest)
```

`as.data.frame.openaq_licenses_list` *Method for converting openaq_licenses_list to data frame.*

Description

Method for converting openaq_licenses_list to data frame.

Usage

```
## S3 method for class 'openaq_licenses_list'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A list of licenses as returned from list_licenses.
`row.names`	`NULL` or a character vector giving the row names for the data frame. Missing values are not allowed.
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
`...`	additional arguments to be passed to or from methods.

Value

A data frame class of the licenses results, with the following columns:

id Numeric. The licenses identifier.

name Character. The license name.

commercial_use_allowed Logical. Indicates whether commerical use is allowed under the license terms.

attribution_required Logical. Indicates whether attribution is required under the license terms.

share_alike_required Logical. Indicates whether share-alike is required under the license terms.

modification_allowed Logical. Indicates whether modification is allowed under the license terms.

redistribution_allowed Logical. Indicates whether redistribution is allowed under the license terms.

source_url String. The URL of the license as listed by the upstream source.

The data frame also includes a `meta` attribute from the original `openaq_licenses_list`.

Examples

```
licenses <- list_licenses(as_data_frame = FALSE)
as.data.frame(licenses)
```

`as.data.frame.openaq_locations_list` *Method for converting openaq_locations_list to data frame.*

Description

Method for converting openaq_locations_list to data frame.

Usage

```
## S3 method for class 'openaq_locations_list'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A list of locations as returned from list_locations.
`row.names`	`NULL` or a character vector giving the row names for the data frame. Missing values are not allowed.
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
`...`	additional arguments to be passed to or from methods.

Value

A data frame class of the locations results, with the following columns:

id Numeric. The locations identifier.

name Character. The name of the location.

is_mobile Logical. Indicates whether the location is stationary or mobile.

is_monitor Logical. Indicates whether the location is considered a reference monitor.

timezone Factor. The IANA timezone of the location (e.g. "America/New_York").

countries_id Numeric. The countries identifier where the location is located.

country_name Character. The name of the country where the location is located

country_iso Factor. The ISO 3166-1 alpha-2 country code where the location is located

latitude Numeric.The latitude, geographic Y, value for the measurement.

longitude Numeric. The longitude, geographic X, value for the measurement.

datetime_first POSIXct. The datetime of the first measurement of this location.

datetime_last POSIXct. The datetime of the last measurement of this location.

owner_name Factor. The name of the owner of the location.

providers_id Numeric. The providers identifier for the location.

provider_name Character. The name of the provider for the location.

The data frame also includes a `meta` attribute from the original `openaq_locations_list`.

Examples

```
loc <- list_locations(as_data_frame = FALSE)
as.data.frame(loc)
```

`as.data.frame.openaq_manufacturers_list`
Method for converting openaq_manufacturers_list to data frame.

Description

Method for converting openaq_manufacturers_list to data frame.

Usage

```
## S3 method for class 'openaq_manufacturers_list'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A list of manufacturers as returned from list_manufacturers.
`row.names`	`NULL` or a character vector giving the row names for the data frame. Missing values are not allowed.
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
`...`	additional arguments to be passed to or from methods.

Value

A data frame class of the manufacturers results, with the following columns:

id Numeric. The manufacturers identifier.

name Character. The name of the manufacturer.

instrument_ids Character. A comma delimited list of instrument identifiers manufactured by the manufacturer.

instrument_names Character. A comma delimited list of instrument names manufactured by this manufacturer.

The data frame also includes a `meta` attribute from the original `openaq_manufacturers_list`.

Examples

```
manufacturers <- list_manufacturers(as_data_frame = FALSE)
as.data.frame(manufacturers)
```

`as.data.frame.openaq_measurements_list` *Method for converting openaq_measurements_list to data frame.*

Description

Method for converting openaq_measurements_list to data frame.

Usage

```
## S3 method for class 'openaq_measurements_list'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A list of measurements as returned from list_sensor_measurements.
`row.names`	`NULL` or a character vector giving the row names for the data frame. Missing values are not allowed.
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
`...`	additional arguments to be passed to or from methods.

Value

A data frame class of the measurements results, with the following columns:

value Numeric. The measurement value.

parameter_id Numeric. The parameter identifier for the measurement.

parameter_name Character. The name of the measured parameter.

parameter_units Character. The units of the measured parameter.

period_label Factor. The label describing the measurement period (e.g. "hour", "day").

period_interval Factor. The period of the measurement interval in HH:MM:SS format (e.g. "01:00:00").

datetime_from POSIXct. The start datetime of the measurement period in local time.

datetime_to POSIXct. The end datetime of the measurement period in local time.

latitude Numeric. The latitude, geographic Y, value for the measurement.

longitude Numeric. The longitude, geographic X, value for the measurement.

min Numeric. The minimum value within the measurement period.

q02 Numeric. The 2nd percentile value within the measurement period.

q25 Numeric. The 25th percentile value within the measurement period.

median Numeric. The median value within the measurement period.

q75 Numeric. The 75th percentile value within the measurement period.

q98 Numeric. The 98th percentile value within the measurement period.

max Numeric. The maximum value within the measurement period.

avg Numeric. The average value within the measurement period.

sd Numeric. The standard deviation of values within the measurement period.

expected_count Numeric. The expected number of measurements within the period.

expected_interval Factor. The expected measurement interval in HH:MM:SS format (e.g. "01:00:00").

observed_count Numeric. The observed number of measurements within the period.

observed_interval Factor. The observer measurement interval in HH:MM:SS format (e.g. "01:00:00").

percent_complete Numeric. The percentage of expected measurements that were observed.

percent_coverage Numeric. The percentage of time coverage for the measurement period.

The data frame also includes a `meta` attribute from the original `openaq_measurements_list`.

Examples

```
meas <- list_sensor_measurements(23707, limit = 500, as_data_frame = FALSE)
as.data.frame(meas)
```

---

as.data.frame.openaq_owners_list

Method for converting openaq_owners_list to data frame.

Description

Method for converting openaq_owners_list to data frame.

Usage

```
## S3 method for class 'openaq_owners_list'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A list of owners as returned from list_owners.
`row.names`	`NULL` or a character vector giving the row names for the data frame. Missing values are not allowed.
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
`...`	additional arguments to be passed to or from methods.

Value

A data frame class of the owners results, with the following columns:

id Numeric. The owners identifier.

name Character. The owners name.

The data frame also includes a `meta` attribute from the original `openaq_owners_list`.

Examples

```
owners <- list_owners(as_data_frame = FALSE)
as.data.frame(owners)
```

---

as.data.frame.openaq_parameters_list

Method for converting openaq_parameters_list to data frame.

Description

Method for converting openaq_parameters_list to data frame.

Usage

```
## S3 method for class 'openaq_parameters_list'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A list of parameters as returned from list_parameters.
`row.names`	`NULL` or a character vector giving the row names for the data frame. Missing values are not allowed.
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
`...`	additional arguments to be passed to or from methods.

Value

A data frame class of the parameters results, with the following columns:

id Numeric. The parameter identifier.

name Character. Name of the parameter.

units Character. The units used by the parameter.

display_name Character. The combined name of the parameter and units. May be NA if not provided.

description Character. Description of the parameter. May be NA if not provided.

The data frame also includes a `meta` attribute from the original `openaq_parameters_list`.

Examples

```
parameters <- list_parameters(as_data_frame = FALSE)
as.data.frame(parameters)
```

---

as.data.frame.openaq_providers_list

Method for converting openaq_providers_list to data frame.

Description

Method for converting openaq_providers_list to data frame.

Usage

```
## S3 method for class 'openaq_providers_list'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A list of providers as returned from list_providers.
`row.names`	`NULL` or a character vector giving the row names for the data frame. Missing values are not allowed.
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
`...`	additional arguments to be passed to or from methods.

Value

A data frame class of the providers results, with the following columns:

id Numeric. The providers identifier.

name Character. The name of the provider.

source_name Factor. The name of the source.

export_prefix Character. Prefixed when exported to file store.

datetime_added POSIXct. Datetime when the provider was first added to OpenAQ.

datetime_first POSIXct. Datetime of the first measurement value from the provider.

datetime_last POSIXct. Datetime of the last measurement value from the provider.

entities_id Numeric. Entities identifier for the provider.

parameter_ids Character. A comma delimited list of parameters identifier measured by the provider.

The data frame also includes a `meta` attribute from the original `openaq_providers_list`.

Examples

```
providers <- list_providers(as_data_frame = FALSE)
as.data.frame(providers)
```

---

as.data.frame.openaq_sensors_list

Method for converting openaq_sensors_list to data frame.

Description

Method for converting openaq_sensors_list to data frame.

Usage

```
## S3 method for class 'openaq_sensors_list'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A list of sensors as returned from get_sensor or list_location_sensors.
`row.names`	`NULL` or a character vector giving the row names for the data frame. Missing values are not allowed.
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
`...`	additional arguments to be passed to or from methods.

Value

A data frame class of the sensors results, with the following columns:

id Numeric. The sensors identifier.

name Character.

parameters_id Numeric. .

datetime_first_utc POSIXct. The datetime of the first measurement in UTC.

datetime_first_local POSIXct. The datetime of the first measurement in local time.

datetime_last_utc POSIXct. The datetime of the last measurement in UTC.

datetime_last_local POSIXct. The datetime of the last measurement in local time.

min Numeric. The minimum measurement value recorded by the sensor.

max Numeric. The maximum measurement value recorded by the sensor.

avg Numeric. The average measurement value recorded by the sensor.

expected_count Numeric. The expected number of measurements for the sensor.

expected_interval Factor. The expected measurement interval in HH:MM:SS format (e.g. "01:00:00").

observed_count Numeric. The observed number of measurements for the sensor.

observed_interval Factor. The observed measurement interval in HH:MM:SS format (e.g. "01:00:00").

percent_complete Numeric. The percentage of expected measurements that were observed.

percent_coverage Numeric. The percentage of time coverage for the sensor.

latest_value Numeric. The most recent measurement value from the sensor.

latest_datetime POSIXct. The datetime of the most recent measurement.

latest_latitude Numeric. The latitude of the most recent measurement location.

latest_longitude Numeric. The longitude of the most recent measurement location

The data frame also includes a `meta` attribute from the original `openaq_sensors_list`.

Examples

```
sensor <- get_sensor(42, as_data_frame = FALSE)
as.data.frame(sensor)
```

`enable_rate_limit` *Toggles on the RATE_LIMIT environment variable to TRUE.*

Description

Toggles on the RATE_LIMIT environment variable to TRUE.

Usage

```
enable_rate_limit()
```

Value

No return value, called for side effects.

Examples

```
enable_rate_limit()
```

get_country *Get a single country from countries resource.*

Description

Get a single country from countries resource.

Usage

```
get_country(
  countries_id,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

countries_id	An integer representing the OpenAQ countries_id.
as_data_frame	A logical for toggling whether to return results as data frame or list, default is TRUE.
dry_run	A logical for toggling a dry run of the request, default is FALSE.
rate_limit	A logical for toggling automatic rate limiting based on rate limit headers, default is FALSE.
api_key	A valid OpenAQ API key string, default is NULL.

Value

A data frame or list of the results.

Examples

```
country <- get_country(42)
```

`get_instrument` *Get a single instrument from the instruments resource.*

Description

Get a single instrument from the instruments resource.

Usage

```
get_instrument(
  instruments_id,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`instruments_id`	An integer representing the OpenAQ instruments_id.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
instrument <- get_instrument(42)
```

`get_license` *Get a single license from the licenses resource.*

Description

Get a single license from the licenses resource.

Usage

```
get_license(
  licenses_id,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`licenses_id`	An integer representing the OpenAQ licenses_id.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
license <- get_license(42)
```

`get_location` *Get a single location from the locations resource.*

Description

Get a single location from the locations resource.

Usage

```
get_location(
  locations_id,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`locations_id`	An integer representing the locations_id to request.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or list of results.

Examples

```
location <- get_location(42)
```

get_manufacturer *Get a single manufacturer from the manufacturers resource.*

Description

Get a single manufacturer from the manufacturers resource.

Usage

```
get_manufacturer(
  manufacturers_id,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

manufacturers_id	An integer representing the OpenAQ manufacturers_id.
as_data_frame	A logical for toggling whether to return results as data frame or list, default is TRUE.
dry_run	A logical for toggling a dry run of the request, default is FALSE.
rate_limit	A logical for toggling automatic rate limiting based on rate limit headers, default is FALSE.
api_key	A valid OpenAQ API key string, default is NULL.

Value

A data frame or a list of the results.

Examples

```
manufacturer <- get_manufacturer(42)
```

`get_owner` *Get a single owner from owners resource.*

Description

Get a single owner from owners resource.

Usage

```
get_owner(
  owners_id,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`owners_id`	An integer representing the OpenAQ owners_id.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
owners <- get_owner(42)
```

get_parameter *Get a single parameter from the parameters resource.*

Description

Get a single parameter from the parameters resource.

Usage

```
get_parameter(
  parameters_id,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

parameters_id	An integer representing the OpenAQ parameters_id.
as_data_frame	A logical for toggling whether to return results as data frame or list, default is TRUE.
dry_run	A logical for toggling a dry run of the request, default is FALSE.
rate_limit	A logical for toggling automatic rate limiting based on rate limit headers, default is FALSE.
api_key	A valid OpenAQ API key string, default is NULL.

Value

A data frame or a list of the results.

Examples

```
parameter <- get_parameter(42)
```

`get_provider` *Get a single provider from providers resource.*

Description

Get a single provider from providers resource.

Usage

```
get_provider(
  providers_id,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`providers_id`	An integer representing the OpenAQ providers_id.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
provider <- get_provider(42)
```

`get_sensor` *Get a single sensor from sensors resource.*

Description

Get a single sensor from sensors resource.

Usage

```
get_sensor(
  sensors_id,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`sensors_id`	An integer representing the OpenAQ sensors_id.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
sensor <- get_sensor(42)
```

`list_countries` *Get a list of countries from the countries resource.*

Description

Get a list of countries from the countries resource.

Usage

```
list_countries(
  providers_id = NULL,
  parameters_id = NULL,
  order_by = NULL,
  sort_order = NULL,
  limit = NULL,
  page = NULL,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`providers_id`	A numeric vector of length 1 or more, containing the ID(s) of the providers to use for filtering results. If multiple IDs are provided, results matching any of the IDs will be returned.
`parameters_id`	A numeric vector of length 1 or more, containing the ID(s) of the parameters to use for filtering results. If multiple IDs are provided, results matching any of the IDs will be returned.
`order_by`	A character string specifying the field to order results by.
`sort_order`	A character string specifying sort direction, either `"asc"` or `"desc"`.
`limit`	An integer specifying the maximum number of results to return, default is `100`.
`page`	An integer specifying the page number for paginated results, default is `1`.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
countries <- list_countries()
```

`list_instruments` *Get a list of instruments from the instruments resource.*

Description

Get a list of instruments from the instruments resource.

Usage

```
list_instruments(
  order_by = NULL,
  sort_order = NULL,
  limit = NULL,
  page = NULL,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`order_by`	A character string specifying the field to order results by.
`sort_order`	A character string specifying sort direction, either `"asc"` or `"desc"`.
`limit`	An integer specifying the maximum number of results to return, default is `100`.
`page`	An integer specifying the page number for paginated results, default is `1`.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
instruments <- list_instruments()
```

`list_licenses` *Get a list of licenses from the licenses resource.*

Description

Get a list of licenses from the licenses resource.

Usage

```
list_licenses(
  order_by = NULL,
  sort_order = NULL,
  limit = NULL,
  page = NULL,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`order_by`	A character string specifying the field to order results by.
`sort_order`	A character string specifying sort direction, either `"asc"` or `"desc"`.
`limit`	An integer specifying the maximum number of results to return, default is `100`.
`page`	An integer specifying the page number for paginated results, default is `1`.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
licenses <- list_licenses()
```

`list_location_latest` *Get the latest measurements by locations_id.*

Description

Get the latest measurements by locations_id.

Usage

```
list_location_latest(
  locations_id,
  datetime_min = NULL,
  limit = NULL,
  page = NULL,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`locations_id`	An integer representing the OpenAQ locations_id.
`datetime_min`	A POSIXct datetime specifying the minimum datetime for filtering results, default is `NULL`.
`limit`	An integer specifying the maximum number of results to return, default is `100`.
`page`	An integer specifying the page number for paginated results, default is `1`.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
measurements <- list_location_latest(2178)
```

`list_location_sensors` *Get a list of a location's sensors.*

Description

Get a list of a location's sensors.

Usage

```
list_location_sensors(
  locations_id,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`locations_id`	An integer representing the OpenAQ locations_id.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
sensors <- list_location_sensors(42)
```

`list_locations` *Get a list of locations from the locations resource.*

Description

Get a list of locations from the locations resource.

Usage

```
list_locations(
  bbox = NULL,
  coordinates = NULL,
  radius = NULL,
  providers_id = NULL,
  parameters_id = NULL,
  owner_contacts_id = NULL,
  manufacturers_id = NULL,
  licenses_id = NULL,
  monitor = NULL,
  mobile = NULL,
  instruments_id = NULL,
  iso = NULL,
  countries_id = NULL,
  order_by = NULL,
  sort_order = NULL,
  limit = NULL,
  page = NULL,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`bbox`	Named numeric vector with four coordinates in form X minimum, Y minimum, X maximum, Y maximum, named values must be `xmin`, `ymin`, `ymax` , `xmax`. default is `NULL`.
`coordinates`	Named numeric vector with two numeric WGS84 (EPSG:4326) geographic coordinates, with named values `latitude` and `longitude`. Represents the central point to be used in conjunction with the radius parameter for geographic search. default is `NULL`.
`radius`	An integer for the number of meters to search around the `coordinates` parameter for filtering locations within the radius. Value must be greater than zero and less than 25000 (25km). default is `NULL`.

providers_id A numeric vector of length 1 or more, containing the ID(s) of the provider(s) to use for filtering results. If multiple IDs are provided, results matching any of the IDs will be returned. default is NULL.

parameters_id A numeric vector of length 1 or more, containing the ID(s) of the parameter(s) to use for filtering results. If multiple IDs are provided, results matching any of the IDs will be returned. default is NULL.

owner_contacts_id A numeric vector of length 1 or more, containing the ID(s) of the owners(s) to use for filtering results. If multiple IDs are provided, results matching any of the IDs will be returned. default is NULL.

manufacturers_id A numeric vector of length 1 or more, containing the ID(s) of the manufacturers(s) to use for filtering results. If multiple IDs are provided, results matching any of the IDs will be returned. default is NULL.

licenses_id A numeric vector of length 1 or more, containing the ID(s) of the license(s) to use for filtering results. If multiple IDs are provided, results matching any of the IDs will be returned. default is NULL.

monitor A logical to filter results to regulatory monitors (TRUE) or air sensors (FALSE), both are included if NULL, default is NULL.

mobile A logical to filter results to mobile (TRUE) or stationary (FALSE) location, both are included if NULL, default is NULL.

instruments_id A numeric vector of length 1 or more, containing the ID(s) of the instrument(s) to use for filtering results. If multiple IDs are provided, results matching any of the IDs will be returned. default is NULL.

iso An ISO 3166-1 alpha-2 string of the country to filter the results, , default is NULL.

countries_id A numeric vector of length 1 or more containing country IDs to use for filtering results. If multiple IDs are provided, results matching any of the IDs will be returned. default is NULL.

order_by A character string specifying the field to order results by.

sort_order A character string specifying sort direction, either "asc" or "desc".

limit An integer specifying the maximum number of results to return, default is 100.

page An integer specifying the page number for paginated results, default is 1.

as_data_frame A logical for toggling whether to return results as data frame or list, default is TRUE.

dry_run A logical for toggling a dry run of the request, default is FALSE.

rate_limit A logical for toggling automatic rate limiting based on rate limit headers. default is FALSE.

api_key A valid OpenAQ API key string. default is NULL.

Value

A data frame or list of results.

Examples

```
locations <- list_locations()
```

`list_manufacturer_instruments` *Get a list of manufacturer instruments from the instruments resource.*

Description

Get a list of manufacturer instruments from the instruments resource.

Usage

```
list_manufacturer_instruments(
  manufacturers_id,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`manufacturers_id`	An integer representing the OpenAQ manufacturers_id.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
instruments <- list_manufacturer_instruments(42)
```

`list_manufacturers` *Get a list of manufacturers from the manufacturers resource.*

Description

Get a list of manufacturers from the manufacturers resource.

Usage

```
list_manufacturers(
  order_by = NULL,
  sort_order = NULL,
  limit = NULL,
  page = NULL,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`order_by`	A character string specifying the field to order results by.
`sort_order`	A character string specifying sort direction, either `"asc"` or `"desc"`.
`limit`	An integer specifying the maximum number of results to return, default is `100`.
`page`	An integer specifying the page number for paginated results, default is `1`.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
manufacturers <- list_manufacturers()
```

`list_owners` *Get a list of owners from the owners resource.*

Description

Get a list of owners from the owners resource.

Usage

```
list_owners(
  order_by = NULL,
  sort_order = NULL,
  limit = NULL,
  page = NULL,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`order_by`	A character string specifying the field to order results by.
`sort_order`	A character string specifying sort direction, either `"asc"` or `"desc"`.
`limit`	An integer specifying the maximum number of results to return, default is `100`.
`page`	An integer specifying the page number for paginated results, default is `1`.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
owners <- list_owners()
```

`list_parameter_latest` *Get the latest measurements by parameters_id.*

Description

Get the latest measurements by parameters_id.

Usage

```
list_parameter_latest(
  parameters_id,
  datetime_min = NULL,
  limit = NULL,
  page = NULL,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`parameters_id`	An integer representing the OpenAQ parameters_id
`datetime_min`	A POSIXct datetime specifying the minimum datetime for filtering results, default is `NULL`.
`limit`	An integer specifying the maximum number of results to return, default is `100`.
`page`	An integer specifying the page number for paginated results, default is `1`.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
measurements <- list_parameter_latest(2)
```

`list_parameters` *Get a list of parameters from the parameters resource.*

Description

Get a list of parameters from the parameters resource.

Usage

```
list_parameters(
  order_by = NULL,
  sort_order = NULL,
  limit = 100,
  page = 1,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`order_by`	A character string specifying the field to order results by.
`sort_order`	A character string specifying sort direction, either `"asc"` or `"desc"`.
`limit`	An integer specifying the maximum number of results to return, default is `100`.
`page`	An integer specifying the page number for paginated results, default is `1`.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
parameters <- list_parameters()
```

`list_providers` *Get a list of providers from the providers resource.*

Description

Get a list of providers from the providers resource.

Usage

```
list_providers(
  order_by = NULL,
  sort_order = NULL,
  limit = 100,
  page = 1,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`order_by`	A character string specifying the field to order results by.
`sort_order`	A character string specifying sort direction, either `"asc"` or `"desc"`.
`limit`	An integer specifying the maximum number of results to return, default is `100`.
`page`	An integer specifying the page number for paginated results, default is `1`.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
providers <- list_providers()
```

`list_sensor_measurements` *Get a list of measurements by sensors_id.*

Description

Get a list of measurements by sensors_id.

Usage

```
list_sensor_measurements(
  sensors_id,
  data = "measurements",
  rollup = NULL,
  datetime_from = NULL,
  datetime_to = NULL,
  order_by = NULL,
  sort_order = NULL,
  limit = NULL,
  page = NULL,
  as_data_frame = TRUE,
  dry_run = FALSE,
  rate_limit = FALSE,
  api_key = NULL
)
```

Arguments

`sensors_id`	An integer representing an OpenAQ sensors_id.
`data`	A character string for the data interval to return, default is "measurements".
`rollup`	A character string representing the aggregation rollup, default is `NULL`.
`datetime_from`	A POSIXct datetime (when `data` is `"measurements"` or `"hours"`) or a Date (when `data` is `"days"` or larger) to filter from, default is `NULL`.
`datetime_to`	A POSIXct datetime (when `data` is `"measurements"` or `"hours"`) or a Date (when `data` is `"days"` or larger) to filter to, default is `NULL`.
`order_by`	A character string representing the field to order by, default is `NULL`.
`sort_order`	A character string, either "asc" or "desc", default is `NULL`.
`limit`	An integer representing the number of results per page.
`page`	An integer representing the page number.
`as_data_frame`	A logical for toggling whether to return results as data frame or list, default is `TRUE`.
`dry_run`	A logical for toggling a dry run of the request, default is `FALSE`.
`rate_limit`	A logical for toggling automatic rate limiting based on rate limit headers, default is `FALSE`.
`api_key`	A valid OpenAQ API key string, default is `NULL`.

Value

A data frame or a list of the results.

Examples

```
measurements <- list_sensor_measurements(3920, "hours")
```

`plot.openaq_locations_data.frame`

Helper for plotting locations on map.

Description

Plots location coordinates. If the `maps` package is installed, a world boundary overlay is added. Install with `install.packages("maps")`.

Usage

```
## S3 method for class 'openaq_locations_data.frame'
plot(x, y = NULL, ...)
```

Arguments

`x`	the coordinates of points in the plot. Alternatively, a single plotting structure, function or any R object with a plot method can be provided.
`y`	the y coordinates of points in the plot, optional if x is an appropriate structure.
`...`	Other options passed on to base::plot().

Value

Called for its side effect of producing a plot. Returns `NULL` invisibly.

Examples

```
df <- list_locations(limit = 100)
plot(df, pch = 19, col = df$provider_name)
```

plot.openaq_locations_list *Helper for plotting locations from list.*

Description

Helper for plotting locations from list.

Usage

```
## S3 method for class 'openaq_locations_list'
plot(x, y = NULL, ...)
```

Arguments

x	A list of locations results.
y	default is NULL
...	Other options passed on to base::plot().

Value

Called for its side effect of producing a plot. Returns NULL invisibly.

Examples

```
loc <- list_locations(limit = 6, as_data_frame = FALSE)
plot(loc, pch = 19, col = 2)
```

plot.openaq_measurements_data.frame *Helper for plotting measurements*

Description

Helper for plotting measurements

Usage

```
## S3 method for class 'openaq_measurements_data.frame'
plot(x, y = NULL, ...)
```

Arguments

x	A data frame of measurements results.
y	Unused, default is NULL.
...	Other options to be passed on to base::plot().

Value

Called for its side effect of producing a plot. Returns NULL invisibly.

Examples

```
meas <- list_sensor_measurements(23707, limit = 500, as_data_frame = FALSE)
plot(meas)
```

---

plot.openaq_measurements_list

Helper for plotting measurements from list

Description

Helper for plotting measurements from list

Usage

```
## S3 method for class 'openaq_measurements_list'
plot(x, y = NULL, ...)
```

Arguments

x	A list of measurements results.
y	Other data
...	Other options to be passed on to base::plot().

Value

Called for its side effect of producing a plot. Returns NULL invisibly.

Examples

```
meas <- list_sensor_measurements(23707, limit = 500)
plot(meas)
```

set_api_key *Set the API key value.*

Description

A helper function to set the OPENAQ_API_KEY environment variable.

Usage

```
set_api_key(api_key)
```

Arguments

api_key	A character string value for the API key to set.

Value

No return value, called for side effects.

Examples

```
set_api_key("my-super-secret-openaq-api-key-1234")
```

set_base_url *Sets base URL environment variable*

Description

A helper function to set the OPENAQR_BASE_URL environment variable. This is to override the default URL for testing and custom instance of the API. This function is generally not used by most users except in extraordinary cases.

Usage

```
set_base_url(base_url)
```

Arguments

base_url	A character string containing a URL.

Value

No return value, called for side effects.

Examples

```
set_base_url("https://example.com")
```

Index