Package ‘airnow’

September 20, 2026

Title Retrieve 'AirNow' Air Quality Observations and Forecasts

Version 0.2.0

Description Retrieve air quality data via the 'AirNow' <https://www.airnow.gov/> API.

License MIT + file LICENSE

URL https://briandconnelly.github.io/airnow/, https://github.com/briandconnelly/airnow

BugReports https://github.com/briandconnelly/airnow/issues

Depends R (>= 4.1.0)

Imports cli, glue, httr2, jsonlite, lifecycle, rlang (>= 1.0.0), tibble

Suggests covr, httptest2, lintr, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Brian Connelly [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-9948-0379>)

Maintainer Brian Connelly <bdc@bconnelly.net>

Repository CRAN

Date/Publication 2026-09-20 02:50:07 UTC

Contents

airnow_areas 2
aqi_color 3
get_airnow_forecasts 4
get_airnow_forecast_history 5

get_airnow_key 6
get_airnow_monitors 7
get_airnow_observations 8
get_airnow_reporting_area 10

Index **12**

`airnow_areas` *AirNow reporting areas*

Description

Metadata for every AirNow reporting area: its location, time zone, agency, and the rules AirNow uses to pick monitors for it. Reporting-area codes are the `area` argument to `get_airnow_observations()`, `get_airnow_forecasts()`, and `get_airnow_forecast_history()`.

Usage

```
airnow_areas
```

Format

A tibble with 1,036 rows and 14 columns:

reporting_area Area name. Not unique: 26 names appear in more than one state (e.g. Aberdeen, SD and Aberdeen, WA).

state_code Two-letter state or province code

country_code Two-letter country code (US, CA, MX)

latitude, longitude Representative point, decimal degrees

gmt_offset Standard-time offset from UTC, in hours. AirNow's source rounds fractional offsets to whole hours; known half- and quarter-hour zones are corrected in this dataset.

observes_dst Whether the area observes daylight saving time

tz_standard, tz_daylight Time zone abbreviations as AirNow records them. Some are non-standard (Anchorage: `AKT`/`ADT`).

reporting_area_code Unique code, e.g. `"ca064"`

agency Agency responsible for the area

lookup_behavior How AirNow picks monitors, e.g. `"Closest Reading By Pollutant"`

considered_monitors Which monitors are eligible

lookup_boundary Search radius, e.g. `"50 miles"`; `NA` when AirNow leaves it blank

Details

Reporting areas are occasionally added, renamed, or retired between package releases, so a live response can name an area missing from this table. Functions that join against it warn when that happens and leave the geography columns `NA`.

Source

AirNow reporting area metadata, https://files.airnowtech.org/airnow/today/reportingarea_metadata.dat, retrieved 2026-09-12. AirNow is a partnership of the U.S. EPA, NOAA, NPS, tribal, state, and local agencies; the file is published without access restrictions. One exact duplicate row (Monterrey, `mx002`) was removed. Rebuild with `data-raw/airnow_areas.R`.

`aqi_color` *Label AQI Values*

Description

`aqi_color()` returns the color that corresponds with the given AQI value.

`aqi_descriptor()` converts the given AQI value(s) into a descriptive string.

Usage

```
aqi_color(aqi)

aqi_descriptor(aqi)
```

Arguments

`aqi`	A vector of AQI values (whole numbers, `NA` allowed)

Details

The AQI scale nominally tops out at 500, but AirNow reports higher values during severe smoke events. Values above 500 are treated as Hazardous. `NA` inputs give `NA` outputs. Negative values (including AirNow's `-1` sentinel for a categorical forecast) give `NA` with a warning.

Value

`aqi_color()` returns a character vector of RGB hex strings

`aqi_descriptor()` returns a character vector

Examples

```
aqi_color(35)
aqi_color(c(35, NA, 874))
aqi_descriptor(35)
```

get_airnow_forecasts *Get current air quality forecasts*

Description

get_airnow_forecasts() retrieves the forecasts currently issued for a reporting area, located by ZIP code, latitude/longitude, or reporting area code. One row is returned per pollutant per forecast day. For forecasts valid on past dates, use get_airnow_forecast_history().

Usage

```
get_airnow_forecasts(
  zip = NULL,
  latitude = NULL,
  longitude = NULL,
  area = NULL,
  clean_names = TRUE,
  api_key = get_airnow_key()
)
```

Arguments

zip	ZIP code, a 5-digit numeric string (e.g., "90210")
latitude	Latitude in decimal degrees
longitude	Longitude in decimal degrees
area	Reporting area code such as "ca064". See airnow_areas or get_airnow_reporting_area(). When given, zip, latitude, and longitude are ignored.
clean_names	Whether column names should be converted to snake_case (default: TRUE). With FALSE, the API's lowerCamelCase names are kept.
api_key	AirNow API key

Value

A tibble with one row per pollutant per forecast day. parameter is a factor with levels ozone, pm2.5, pm10, co, no2, so2; category_name is an ordered factor from Good to Hazardous. latitude and longitude describe the reporting area and come from airnow_areas. An aqi of -1 means the agency issued a categorical forecast only.

Requests made

One request per call. Every request counts against AirNow's limit of 500 requests per hour per key.

See Also

get_airnow_forecast_history() for past forecasts.

Examples

```
## Not run:
get_airnow_forecasts(zip = "90210")
get_airnow_forecasts(area = "ca064")

## End(Not run)
```

---

get_airnow_forecast_history

Get past air quality forecasts

Description

`get_airnow_forecast_history()` retrieves forecasts that were *valid* on dates in the given range for one reporting area. Every forecast lead time is returned unless `range` narrows it: `range = 1` keeps only the forecast issued the day before each valid date.

Usage

```
get_airnow_forecast_history(
  area,
  start_date,
  end_date,
  range = NULL,
  parameter = NULL,
  clean_names = TRUE,
  api_key = get_airnow_key()
)
```

Arguments

`area`	Reporting area code such as `"ca064"` (required). See airnow_areas or `get_airnow_reporting_area()`.
`start_date, end_date`	First and last *valid* date to include, as `Date` objects or `"YYYY-MM-DD"` strings. AirNow serves these forecasts only through today, so a future date returns no rows and warns. Use `get_airnow_forecasts()` for upcoming forecasts.
`range`	Optional forecast lead time in days to keep (a positive whole number). `NULL` (default) keeps every lead time.
`parameter`	Optional pollutant to keep: one of `"ozone"`, `"pm2.5"`, `"pm10"`, `"co"`, `"no2"`, `"so2"`.
`clean_names`	Whether column names should be converted to snake_case (default: `TRUE`). With `FALSE`, the API's lowerCamelCase names are kept.
`api_key`	AirNow API key

Value

A tibble with the same columns as `get_airnow_forecasts()`.

Requests made

One request per call. Every request counts against AirNow's limit of 500 requests per hour per key.

See Also

`get_airnow_reporting_area()` to find the area code for a ZIP code or coordinates; `get_airnow_forecasts()` for current forecasts.

Examples

```
## Not run:
get_airnow_forecast_history("md008", "2026-01-13", "2026-01-14")
get_airnow_forecast_history("md008", "2026-01-13", "2026-01-13",
  range = 1, parameter = "pm2.5"
)

# Starting from a ZIP code
area <- get_airnow_reporting_area(zip = "90210")$reporting_area_code
get_airnow_forecast_history(area, "2026-09-01", "2026-09-03", range = 1)

## End(Not run)
```

---

get_airnow_key *Manage your AirNow API key*

Description

`get_airnow_key()` returns the configured AirNow API key. If no key is set and `ask` is `TRUE` in an interactive session, you are prompted for one; otherwise an error is raised.

The key is read from the `AIRNOW_API_KEY` environment variable. To set it permanently, add `AIRNOW_API_KEY=your-key` to your `~/.Renviron` file. Keys are issued at https://docs.airnowapi.org/account/request/.

`set_airnow_key()` sets the AirNow API key for the current session.

Usage

```
get_airnow_key(ask = is_interactive())

set_airnow_key(key = NULL, ask = is_interactive())
```

Arguments

`ask`	Whether to prompt for the key if none is set. Prompting only works in interactive sessions.
`key`	The API key to use. If `NULL` and `ask` is `TRUE`, you are prompted for one.

Value

`get_airnow_key()` returns a string.

`set_airnow_key()` returns the key, invisibly.

Examples

```
## Not run:
get_airnow_key()

## End(Not run)
## Not run:
set_airnow_key(key = "4d36e978-e325-11ce-bfc1-08002be10318")

## End(Not run)
```

---

get_airnow_monitors *Get air quality data from monitoring sites in a region*

Description

`get_airnow_monitors()` retrieves readings from every monitoring site inside a bounding box. Before airnow 0.2.0 this function was called `get_airnow_area()`.

Usage

```
get_airnow_monitors(
  box,
  parameters = "pm25",
  start_time = NULL,
  end_time = NULL,
  monitor_type = "both",
  data_type = c("aqi", "concentrations", "both"),
  verbose = FALSE,
  raw_concentrations = FALSE,
  clean_names = TRUE,
  api_key = get_airnow_key()
)
```

Arguments

box	Four-element numeric vector specifying a bounding box for the region of interest. Format is (minX, minY, maxX, maxY), where X and Y are longitude and latitude, respectively.
parameters	Parameter(s) to return data for. Choices are PM_{2.5} (`pm25`: default), `ozone`, PM_10 (`pm10`), CO (`co`), NO2 (`no2`), and SO2 (`so2`).

`start_time` Optional. The date and time (UTC) at the start of the time period requested. If specified, `end_time` must also be given. If not specified, the most recent past hour is used.

`end_time` Optional. The date and time (UTC) at the end of the time period requested. If specified, `start_time` must also be given. If not specified, the following hour is used.

`monitor_type` Type of monitor to be returned, either `"permanent"`, `"mobile"`, or `"both"` (default).

`data_type` Type of data to be returned, either `"aqi"` (default), `"concentrations"`, or `"both"`.

`verbose` Logical value indicating whether or not to include additional site information including Site Name, Agency Name, AQS ID, and Full AQS ID (default: `FALSE`)

`raw_concentrations` Logical value indicating whether or not raw hourly concentration data should be included (default: `FALSE`)

`clean_names` Whether column names should be converted to snake_case (default: `TRUE`). With `FALSE`, the API's PascalCase names are kept.

`api_key` AirNow API key

Value

A data frame with current air quality conditions

Examples

```
## Not run:
# Get air quality data around Washington state
get_airnow_monitors(box = c(-125.394211, 45.295897, -116.736984, 49.172497))

## End(Not run)
```

`get_airnow_observations`

Get current air quality observations

Description

`get_airnow_observations()` retrieves the most recent hourly readings for a location. Locate it by ZIP code or latitude/longitude, or give a reporting area code. The two forms use different AirNow services with different methodologies:

Usage

```
get_airnow_observations(
  zip = NULL,
  latitude = NULL,
  longitude = NULL,
  area = NULL,
  clean_names = TRUE,
  api_key = get_airnow_key()
)
```

Arguments

`zip`	ZIP code, a 5-digit numeric string (e.g., `"90210"`)
`latitude`	Latitude in decimal degrees
`longitude`	Longitude in decimal degrees
`area`	Reporting area code such as `"ca064"`. See airnow_areas or `get_airnow_reporting_area()`. When given, `zip`, `latitude`, and `longitude` are ignored.
`clean_names`	Whether column names should be converted to snake_case (default: `TRUE`). With `FALSE`, the API's lowerCamelCase names are kept.
`api_key`	AirNow API key

Details

- By ZIP code or coordinates, AirNow returns the closest reading for each pollutant within the reporting area's search radius, and names the site it came from (`site_id`, `site_name`).
- By `area`, AirNow returns the area's official value for each pollutant and omits the site columns, which are `NA`.

The `source` column records which service produced each row.

Value

A tibble with one row per pollutant. `parameter` is a factor with levels `ozone`, `pm2.5`, `pm10`, `co`, `no2`, `so2`; `category_name` is an ordered factor; `aqi` is the NowCast AQI.

Requests made

One request per call, plus one extra request (cached for the session) when a ZIP code's reporting area shares its name with an area in another state. There are 26 such names; see airnow_areas. All requests count against AirNow's 500-per-hour limit.

Time columns

`hour_observed` uses AirNow's convention of labeling an hour by its **end**: `18` means the period 17:00-17:59. `date_observed` and `hour_observed` are local to each reporting area, and `local_time_zone` is an abbreviation R cannot interpret. `utc_datetime` is derived from the reporting area's metadata in airnow_areas and marks the **end** of the observation hour (the API's label). `get_airnow_monitors()` labels the same hour by its **start** in `datetime_observed`, so to join the two subtract one hour from

utc_datetime (or add one hour to datetime_observed). utc_datetime is NA when the area or its time zone cannot be matched.

Examples

```
## Not run:
get_airnow_observations(zip = "90210")
get_airnow_observations(latitude = 38.3191, longitude = -122.2998)
get_airnow_observations(area = "ca064")

## End(Not run)
```

---

get_airnow_reporting_area

Find the AirNow reporting area for a location

Description

get_airnow_reporting_area() looks up which AirNow reporting area serves a ZIP code or a latitude/longitude pair. The returned reporting_area_code is the area argument for get_airnow_observations(), get_airnow_forecasts(), and get_airnow_forecast_history(). Resolution uses AirNow's current-forecast service. When the area has no forecast issued today, a ZIP code is looked up in AirNow's bundled ZIP-to-area crosswalk instead; coordinates cannot be resolved that way, so choose a code from airnow_areas.

Usage

```
get_airnow_reporting_area(
  zip = NULL,
  latitude = NULL,
  longitude = NULL,
  api_key = get_airnow_key()
)
```

Arguments

zip	ZIP code, a 5-digit numeric string (e.g., "90210")
latitude	Latitude in decimal degrees
longitude	Longitude in decimal degrees
api_key	AirNow API key

Value

A one-row tibble with columns reporting_area_code, reporting_area, state_code, latitude, and longitude.

Requests made

One request per distinct location per session; the result is cached in memory, so repeated calls for the same location are free. Every request counts against AirNow's limit of 500 requests per hour per key.

Examples

```
## Not run:
get_airnow_reporting_area(zip = "90210")
get_airnow_reporting_area(latitude = 38.3191, longitude = -122.2998)

## End(Not run)
```

Index

∗ **datasets**
    airnow_areas, 2

airnow_areas, 2, *4, 5*, *9, 10*
aqi_color, 3
aqi_descriptor *(*aqi_color*)*, 3

get_airnow_forecast_history, 5
get_airnow_forecast_history(), *2*, *4*, *10*
get_airnow_forecasts, 4
get_airnow_forecasts(), *2*, *5, 6*, *10*
get_airnow_key, 6
get_airnow_monitors, 7
get_airnow_monitors(), *9*
get_airnow_observations, 8
get_airnow_observations(), *2*, *10*
get_airnow_reporting_area, 10
get_airnow_reporting_area(), *4–6*, *9*

set_airnow_key *(*get_airnow_key*)*, 6