Package: nycOpenData (via r-universe)

November 2, 2026

Type Package

Title A Lightweight Interface to NYC Open Data APIs

Version 0.2.3

Description Provides a unified set of helper functions to access datasets from the NYC Open Data platform <https://opendata.cityofnewyork.us/>. Functions return results as tidy tibbles and support optional filtering, sorting, and row limits via the Socrata API. The package includes endpoints for 311 service requests, DOB job applications, juvenile justice metrics, school safety, environmental data, event permitting, and additional citywide datasets.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports httr, jsonlite, tibble, janitor, curl, dplyr, rlang

Suggests ggplot2, knitr, rmarkdown, scales, testthat (>= 3.0.0), tidyr, vcr (>= 0.6.0), webmockr

URL https://docs.ropensci.org/nycOpenData/, https://github.com/ropensci/nycOpenData

BugReports https://github.com/ropensci/nycOpenData/issues

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 3.5)

Config/pak/sysreqs libicu-dev libssl-dev

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

Date/Publication 2026-11-02 17:01:56 UTC

RemoteUrl https://github.com/ropensci/nycOpenData

RemoteRef HEAD

RemoteSha 061e56554fd96fe400b745a0c58c43301f92ff40

Contents

nyc_any_dataset 2
nyc_list_datasets 3
nyc_pull_dataset 4

Index **7**

`nyc_any_dataset` *Pull any NYC Open Data dataset from a Socrata JSON endpoint*

Description

Downloads data from any NYC Open Data Socrata JSON endpoint and returns the result as a tibble. This function is useful for datasets that are not included in the curated catalog returned by [nyc_list_datasets()].

Usage

```
nyc_any_dataset(
  json_link,
  limit = 10000,
  timeout_sec = 30,
  clean_names = TRUE,
  coerce_types = TRUE
)
```

Arguments

`json_link`	A single Socrata dataset JSON endpoint URL, such as ‘"https://data.cityofnewyork.us/resource/erm2-nwe9.json"‘.
`limit`	Number of rows to retrieve. Defaults to 10,000.
`timeout_sec`	Request timeout in seconds. Defaults to 30.
`clean_names`	Logical. If ‘TRUE‘, column names are converted to snake_case using [janitor::clean_names()]. Defaults to ‘TRUE‘.
`coerce_types`	Logical. If ‘TRUE‘, the package attempts lightweight, heuristic-based type coercion after downloading the data. Columns are converted only when at least 95 percent of non-missing values can be parsed as the target type. This helps avoid unsafe conversions when source data are inconsistent.

Details

NYC Open Data datasets have Socrata JSON endpoints that usually follow this pattern:

‘https://data.cityofnewyork.us/resource/<dataset_uid>.json‘

For example, the 311 Service Requests dataset has the Socrata UID ‘"erm2-nwe9"‘, so its JSON endpoint is:

‘https://data.cityofnewyork.us/resource/erm2-nwe9.json‘

Users can find a dataset’s UID from the NYC Open Data Portal URL, the API documentation page for the dataset, or the output of [nyc_list_datasets()].

‘nyc_any_dataset()‘ bypasses the package catalog and sends a request directly to the supplied JSON endpoint. Unlike [nyc_pull_dataset()], it does not look up defaults such as catalog keys, date fields, or default ordering.

This function is intended for direct endpoint access. For catalog-based workflows using readable keys or Socrata UIDs, use [nyc_pull_dataset()].

Value

A tibble containing rows from the requested NYC Open Data endpoint.

Examples

```
# Examples that hit the live NYC Open Data API are guarded so CRAN checks
# do not fail when the network is unavailable or slow.
if (interactive() && curl::has_internet()) {
  # Build a JSON endpoint from a Socrata UID
  uid <- "erm2-nwe9"
  endpoint <- paste0("https://data.cityofnewyork.us/resource/", uid, ".json")

  out <- try(nyc_any_dataset(endpoint, limit = 3), silent = TRUE)
  if (!inherits(out, "try-error")) {
    head(out)
  }
}
```

---

nyc_list_datasets *List available NYC Open Data datasets*

Description

Retrieves a live catalog of NYC Open Data datasets from the NYC Open Data Portal catalog table (‘5tqd-u88y‘) and returns dataset metadata that can be used with [nyc_pull_dataset()].

Usage

```
nyc_list_datasets()
```

Details

The returned tibble includes both the official Socrata dataset ‘uid‘ and a human-readable ‘key‘. The ‘uid‘ is the stable identifier used by the NYC Open Data Portal and Socrata API, while the ‘key‘ is generated from the dataset name using [janitor::make_clean_names()] to make datasets easier to reference in R code, especially in classroom and reproducible research settings.

Most users will begin by calling ‘nyc_list_datasets()‘, searching the returned catalog for a dataset of interest, and then passing either the ‘key‘ or ‘uid‘ to [nyc_pull_dataset()].

Value

A tibble containing available NYC Open Data datasets. Important columns include:

key A human-readable dataset key generated from the dataset name.

uid The official Socrata dataset identifier used by NYC Open Data.

name The dataset name from the NYC Open Data catalog.

Additional metadata columns from the NYC Open Data catalog may also be returned.

Examples

```
if (interactive() && curl::has_internet()) {
  catalog <- nyc_list_datasets()

  # View available datasets
  catalog

  # Search for 311-related datasets
  catalog[grepl("311", catalog$name, ignore.case = TRUE), c("key", "uid", "name")]
}
```

---

nyc_pull_dataset *Pull a NYC Open Data dataset*

Description

Downloads a dataset from the NYC Open Data Socrata API using either a human-readable catalog ‘key‘ or the official Socrata dataset ‘uid‘ returned by [nyc_list_datasets()].

Usage

```
nyc_pull_dataset(
  dataset,
  limit = 10000,
  filters = list(),
  date = NULL,
  from = NULL,
  to = NULL,
  date_field = NULL,
  where = NULL,
  order = NULL,
  timeout_sec = 30,
  clean_names = TRUE,
  coerce_types = TRUE
)
```

Arguments

dataset	A single dataset ‘key‘ or Socrata dataset ‘uid‘ from [nyc_list_datasets()]. For example, a key may look like ‘"x311_service_requests_from_2020_to_present"‘, while a UID may look like ‘"erm2-nwe9"‘.
limit	Number of rows to retrieve. Defaults to 10,000.
filters	Optional named list of exact-match filters. Each list name should be a field name in the dataset, and each value should be the value or values to match. Vector values are translated into SQL-style ‘IN‘ conditions in the generated SoQL query. For example, ‘filters = list(borough = c("BROOKLYN", "QUEENS"))‘ returns rows where ‘borough‘ is either ‘"BROOKLYN"‘ or ‘"QUEENS"‘.
date	Optional single date used to match all records from that day. Requires ‘date_field‘.
from	Optional start date, inclusive. Requires ‘date_field‘.
to	Optional end date, exclusive. Requires ‘date_field‘.
date_field	Optional date or datetime column to use with ‘date‘, ‘from‘, or ‘to‘. This must be supplied when any date filter is used. Users can identify available date columns by inspecting the dataset on the NYC Open Data Portal or by pulling a small sample with ‘limit‘.
where	Optional raw SoQL ‘WHERE‘ clause for advanced filtering. SoQL is the Socrata Query Language used by NYC Open Data. If ‘date‘, ‘from‘, or ‘to‘ are also supplied, their generated conditions are combined with ‘where‘ using ‘AND‘.
order	Optional raw SoQL ‘ORDER BY‘ clause, such as ‘"created_date DESC"‘.
timeout_sec	Request timeout in seconds. Defaults to 30.
clean_names	Logical. If ‘TRUE‘, column names are converted to snake_case using [janitor::clean_names()]. Defaults to ‘TRUE‘.
coerce_types	Logical. If ‘TRUE‘, the package attempts lightweight, heuristic-based type coercion after downloading the data. Columns are converted only when at least 95 percent of non-missing values can be parsed as the target type. This helps avoid unsafe conversions when source data are inconsistent.

Details

When a catalog ‘key‘ is supplied, ‘nyc_pull_dataset()‘ first retrieves the live NYC Open Data catalog to look up the corresponding Socrata ‘uid‘, then sends a second request to download the dataset itself. Supplying a ‘uid‘ directly is more stable and avoids ambiguity, while keys are provided for readability and classroom-friendly workflows.

Dataset keys are generated from dataset names using [janitor::make_clean_names()]. Because keys are derived from live catalog metadata, Socrata UIDs are the most stable identifiers.

‘nyc_pull_dataset()‘ is designed for common catalog-based workflows. For arbitrary Socrata JSON endpoints that are not included in the package catalog, use [nyc_any_dataset()].

The ‘filters‘ argument is intended for simple exact-match filtering. For more complex conditions, use the ‘where‘ argument with raw SoQL syntax.

Internally, filter field names are wrapped in ‘TRIM()‘ when constructing SoQL queries to reduce mismatches caused by leading or trailing whitespace in source data.

Type coercion is intentionally conservative. When 'coerce_types = TRUE', the package attempts to infer common R column types from the API response, but columns with inconsistent values may remain character columns.

Datetime coercion is also conservative. Timezone offsets and sub-second precision may not always be preserved during automatic parsing, and columns with inconsistent datetime formats may remain character columns.

Value

A tibble containing rows from the requested NYC Open Data dataset.

Examples

```
if (interactive() && curl::has_internet()) {
  # Pull by human-readable key
  nyc_pull_dataset("x311_service_requests_from_2020_to_present", limit = 3)

  # Pull by Socrata UID
  nyc_pull_dataset("erm2-nwe9", limit = 3)

  # Filter to one borough
  nyc_pull_dataset(
    "erm2-nwe9",
    limit = 3,
    filters = list(borough = "QUEENS")
  )

  # Filter to multiple boroughs
  nyc_pull_dataset(
    "erm2-nwe9",
    limit = 10,
    filters = list(borough = c("BROOKLYN", "QUEENS"))
  )

  # Date filtering
  nyc_pull_dataset(
    "erm2-nwe9",
    from = "2023-01-01",
    to = "2024-01-01",
    date_field = "created_date",
    limit = 100
  )

  # Advanced filtering with raw SoQL
  nyc_pull_dataset(
    "erm2-nwe9",
    where = "borough = 'BROOKLYN' AND agency = 'NYPD'",
    order = "created_date DESC",
    limit = 100
  )
}
```

Index

nyc_any_dataset, 2
nyc_list_datasets, 3
nyc_pull_dataset, 4