

Package: ctOpenData (via r-universe)

October 9, 2026

Type Package

Title A Lightweight Interface to Connecticut Open Data APIs

Version 0.1.1

Description Provides a unified set of helper functions to access datasets from the Connecticut Open Data platform [<https://data.ct.gov/>](https://data.ct.gov/). Functions return results as tidy tibbles and support optional filtering, sorting, and row limits via the Socrata API. The package provides a consistent interface for discovering and downloading datasets from the Connecticut Open Data Portal using human-readable dataset keys or official Socrata dataset identifiers.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 8.0.0

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

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

URL <https://github.com/gomes-sh/ctOpenData>,
<https://nyc-open-data-lab.github.io/ctOpenData/>

BugReports <https://github.com/gomes-sh/ctOpenData/issues>

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 3.5)

Config/pak/sysreqs libicu-dev libssl-dev

Repository <https://nyc-open-data-lab.r-universe.dev>

Date/Publication 2026-09-09 13:32:30 UTC

RemoteUrl <https://github.com/nyc-open-data-lab/ctopendata>

RemoteRef HEAD

RemoteSha 5c20cd112ecacae2163e72e6fff053234c67918a

Contents

ct_any_dataset	2
ct_list_datasets	3
ct_pull_dataset	4

Index	8
--------------	----------

ct_any_dataset	<i>Pull any Connecticut Open Data dataset from a ‘Socrata’ JSON endpoint</i>
----------------	--

Description

Downloads data from any Connecticut 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 [ct_list_datasets()].

Usage

```
ct_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.ct.gov/resource/ffju-s5c5.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

Connecticut Open Data datasets have ‘Socrata’ JSON endpoints that usually follow this pattern:

‘https://data.ct.gov/resource/<dataset_uid>.json’

For example, a dataset with the ‘Socrata’ UID “ffju-s5c5” would have the JSON endpoint:

```
'https://data.ct.gov/resource/ffju-s5c5.json'
```

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

'ct_any_dataset()' bypasses the package catalog and sends a request directly to the supplied JSON endpoint. Unlike [ct_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 [ct_pull_dataset()].

Value

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

Examples

```
# Examples that hit the live Connecticut 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 <- "ffju-s5c5"
  endpoint <- paste0("https://data.ct.gov/resource/", uid, ".json")

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

ct_list_datasets

List available Connecticut Open Data datasets

Description

Retrieves a live catalog of Connecticut Open Data datasets from the Connecticut Open Data Portal catalog table ('CATALOG_DATASET_ID') and returns dataset metadata that can be used with [ct_pull_dataset()].

Usage

```
ct_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 Connecticut 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 ‘`ct_list_datasets()`’, searching the returned catalog for a dataset of interest, and then passing either the ‘key’ or ‘uid’ to `[ct_pull_dataset()]`.

Value

A tibble containing available Connecticut 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 Connecticut Open Data.

name The dataset name from the Connecticut Open Data catalog.

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

Examples

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

  # View available datasets
  catalog

  # Search for datasets containing a keyword
  catalog[
    grepl("spill", catalog$name, ignore.case = TRUE),
    c("key", "uid", "name")
  ]
}
```

ct_pull_dataset

Pull a Connecticut Open Data dataset

Description

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

Usage

```
ct_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 [ct_list_datasets()]. For example, a key may look like “spill_incidents_from_july_1_2022_to_recent_for_download”, while a UID may look like “ffju-s5c5”.
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(FILTER_FIELD = c("VALUE_1", "VALUE_2"))’ returns rows where ‘FILTER_FIELD’ is either “VALUE_1” or “VALUE_2”.
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 Connecticut 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 Connecticut 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 “DATE_FIELD 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, ‘ct_pull_dataset()’ first retrieves the live Connecticut 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.

‘ct_pull_dataset()’ is designed for common catalog-based workflows. For arbitrary ‘Socrata’ JSON endpoints that are not included in the package catalog, use [ct_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 Connecticut Open Data dataset.

Examples

```
if (interactive() && curl::has_internet()) {
  # Pull by human-readable key
  ct_pull_dataset("spill_incidents_from_july_1_2022_to_recent_for_download", limit = 3)

  # Pull by `Socrata` UID
  ct_pull_dataset("ffju-s5c5", limit = 3)

  # Filter to one value
  ct_pull_dataset(
    "ffju-s5c5",
    limit = 3,
    filters = list(incident_type = "Petroleum Incident")
  )

  # Filter to multiple values
  ct_pull_dataset(
    "ffju-s5c5",
    limit = 10,
    filters = list(incident_type = c("Biomedical Incident", "Dielectric Fluid Incident"))
  )

  # Date filtering
```

```
ct_pull_dataset(  
  "ffju-s5c5",  
  from = "2023-01-01",  
  to = "2024-01-01",  
  date_field = "reported_date",  
  limit = 100  
)  
  
# Advanced filtering with raw SoQL  
ct_pull_dataset(  
  "ffju-s5c5",  
  where = "incident_type = 'Petroleum Incident' AND township = 'New Haven'",  
  order = "reported_date DESC",  
  limit = 100  
)  
}
```

Index

`ct_any_dataset`, [2](#)
`ct_list_datasets`, [3](#)
`ct_pull_dataset`, [4](#)