

Package: socratadata (via r-universe)

May 29, 2026

Title Explore Socrata Data with Ease

Version 0.1.1.9001

Description Provides an interface to search, read, query, and retrieve metadata for datasets hosted on 'Socrata' open data portals. Supports all 'Socrata' data types, including spatial data returned as 'sf' objects.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/rextendr/version 0.5.0

SystemRequirements Cargo (Rust's package manager), rustc >= 1.65.0, xz

Depends R (>= 4.2)

Imports cli, httr2, rlang (>= 1.1.0), sf, tibble

Suggests glue, httptest2, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://ryanzomorrodi.github.io/socratadata/>,
<https://github.com/ryanzomorrodi/socratadata>

BugReports <https://github.com/ryanzomorrodi/socratadata/issues>

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev xz-utils libclang-dev

Repository <https://ryanzomorrodi.r-universe.dev>

Date/Publication 2026-05-10 23:35:14 UTC

RemoteUrl <https://github.com/ryanzomorrodi/socratadata>

RemoteRef HEAD

RemoteSha 634377571a9046a6981eaa142b502af911ce6431

Contents

soc_discover	2
soc_metadata	4
soc_query	6
soc_read	7

Index	9
--------------	----------

soc_discover	<i>Discover datasets and public data assets using the Socrata Discovery API</i>
--------------	---

Description

Provides access to the Socrata Discovery API, allowing you to search tens of thousands of government datasets and assets published on the Socrata platform. Governments at all levels publish data on topics including crime, permits, finance, healthcare, research, and performance.

Usage

```
soc_discover(
  attribution = NULL,
  categories = NULL,
  domain_category = NULL,
  domains = NULL,
  ids = NULL,
  names = NULL,
  only = "dataset",
  provenance = NULL,
  query = NULL,
  tags = NULL,
  domain_tags = NULL,
  location = "us",
  limit = 10000
)
```

Arguments

attribution	string; Filter by the attribution or publisher
categories	character vector; Filter by categories.
domain_category	string; Filter by domain category (requires a specified domain).
domains	character vector; Filter to domains.
ids	character vector; Filter by an asset IDs.
names	character vector; Filter by asset names.

only	character vector; Filter to specific asset types. Must be one or more of: "chart", "dataset", "filter", "link", "map", "measure", "story", "system_dataset", "visualization". Default is "dataset".
provenance	string; Filter by provenance: "official" or "community".
query	character string; Filter using a a token matching one from an asset's name, description, category, tags, column names, column fieldnames, column descriptions or attribution.
tags	character vector; Filter by tags associated with the assets.
domain_tags	string; Filter by domain tags associated with the assets (requires a specified domain).
location	string; Regional API domain: "us" (default) or "eu".
limit	whole number; Maximum number of results (cannot exceed 10,000).

Value

A tibble containing metadata for each discovered asset. Columns include:

id Asset identifier (four-by-four ID).

parent_ids Asset parent identifiers.

name Asset name.

attribution Attribution or publisher of the asset.

attribution_link Link to attribution.

contact_email Email to contact asset owner.

resource_type Type of resource: api, calendar, chart, dataset, federated_href, file, filter, form, href, link, map, measure, story, visualization.

owner Owner:

id Owner ID.

display_name Display name of owner.

creator Creator:

id Creator ID.

display_name Display name of creator.

provenance Provenance of asset (official or community).

description Textual description of the asset.

created Date asset was created.

updated Date asset was last updated.

published Date asset was published (if published).

data_last_updated Date asset data was last updated

metadata_last_updated Date asset metadata was last updated

categories Category labels assigned to the asset.

tags Tags associated with the asset.

domain_category Category label assigned by the domain.

domain_tags Tags applied by the domain.

domain_metadata Metadata associated with the asset assigned by the domain.

columns A dataframe with the following columns:

name Names of asset columns.

label Labels of asset columns.

description Description of asset columns.

datatype Datatypes of asset columns.

permalink Permanent URL where the asset can be accessed.

link Direct asset link.

domain Domain of the asset.

license License associated with the asset.

page_views_last_week Page views in the last week.

page_views_last_month Page views in the last month.

page_views_total Total page views.

downloads Total number of downloads.

See Also

<https://dev.socrata.com/docs/other/discovery>

Examples

```
# Search for crime-related datasets in the Public Safety category
results <- soc_discover(
  query = "crime",
  categories = "Public Safety",
  only = "dataset"
)
```

soc_metadata

Extract Socrata Dataset Metadata

Description

Retrieves metadata attributes from a tibble returned by `soc_read()` or using the dataset url, including dataset-level information and column-level descriptions.

Usage

```
soc_metadata(dataset)
```

Arguments

`dataset` A tibble returned by `soc_read()` or a url.

Details

This function pulls out descriptive metadata such as the dataset's ID, title, attribution, category, creation and update timestamps, description, any domain-specific fields, and field descriptions defined by the data provider.

Value

An object of class `soc_meta`, which includes:

id Asset identifier (four-by-four ID).

name Asset name.

attribution Attribution or publisher of the asset.

attribution_link Link to attribution.

resource_type Type of resource: api, calendar, chart, dataset, federated_href, file, filter, form, href, link, map, measure, story, visualization.

owner Owner:

id Owner ID.

display_name Display name of owner.

provenance Provenance of asset (official or community).

description Textual description of the asset.

created Date asset was created.

published Date asset was published (if published).

data_last_updated Date asset data was last updated

metadata_last_updated Date asset metadata was last updated

domain_category Category label assigned by the domain.

domain_tags Tags applied by the domain.

domain_metadata Metadata associated with the asset assigned by the domain.

columns A dataframe with the following columns:

name Names of asset columns.

label Labels of asset columns.

description Description of asset columns.

datatype Datatypes of asset columns.

permalink Permanent URL where the asset can be accessed.

license License associated with the asset.

Examples

```
url <- "https://soda.demo.socrata.com/dataset/USGS-Earthquakes-2012-11-08/3wfw-mdbc/"
data <- soc_read(url, soc_query(limit = 1000L))
metadata <- soc_metadata(data)
print(metadata)
```

```
metadata <- soc_metadata(url)
print(metadata)
```

`soc_query`*Build a Socrata Query Object*

Description

Constructs a structured representation of a Socrata Query Language (SOQL) query that can be used with Socrata API endpoints. This function does not execute the query; it creates an object that can be passed to request functions or printed for inspection.

Usage

```
soc_query(  
  select = "*",  
  where = NULL,  
  group_by = NULL,  
  having = NULL,  
  order_by = NULL,  
  limit = NULL  
)
```

Arguments

<code>select</code>	string; Columns to retrieve.
<code>where</code>	string; Filter conditions.
<code>group_by</code>	string; Fields to group by.
<code>having</code>	string; Conditions to apply to grouped records.
<code>order_by</code>	string; Sort order.
<code>limit</code>	whole number; The maximum number of records to return.

Value

An object of class `soc_query`, which prints in a readable format and can be used to build query URLs.

See Also

Use this with a function that executes Socrata requests, e.g., `soc_read(url, query = soc_query(...))`

Examples

```
query <- soc_query(  
  select = "region, avg(magnitude) as avg_magnitude, count(*) as count",  
  group_by = "region",  
  having = "count >= 5",  
  order_by = "avg_magnitude DESC"  
)  
print(query)
```

```
earthquakes_by_region <- soc_read(
  "https://soda.demo.socrata.com/dataset/USGS-Earthquakes-2012-11-08/3wfw-mdbc/",
  query = query
)
```

soc_read

Read a Socrata Dataset into R

Description

Downloads and parses a dataset from a Socrata open data portal URL, returning it as a tibble or sf object. Metadata is also returned as attributes on the returned object.

Usage

```
soc_read(
  url,
  query = soc_query(),
  alias = "label",
  page_size = 10000,
  include_synthetic_cols = TRUE,
  api_key_id = NULL,
  api_key_secret = NULL,
  timezone = Sys.timezone()
)
```

Arguments

url	string; URL of the Socrata dataset.
query	string or soc_query(); Query parameters specification
alias	string; Use of field alias values. There are three options: <ul style="list-style-type: none"> • "label": field alias values are assigned as a label attribute for each field. • "replace": field alias values replace existing column names. • "drop": field alias values replace existing column names.
page_size	whole number; Maximum number of rows returned per request.
include_synthetic_cols	logical; Should synthetic columns be included?
api_key_id	string; API key ID to authenticate requests. (Can also be stored as "soc_api_key_id" environment variable)
api_key_secret	string; API key secret to authenticate requests. (Can also be stored as "soc_api_key_secret" environment variable)
timezone	string; Timezone to set floating_timestamps to.

Value

A tibble with an additional `soc_meta` attribute storing metadata. If the dataset contains a single non-nested geospatial field, it will be returned as an `sf` object.

Examples

```
soc_read(  
  "https://soda.demo.socrata.com/dataset/USGS-Earthquakes-2012-11-08/3wfw-mdbc/"  
)  
  
soc_read(  
  "https://soda.demo.socrata.com/dataset/USGS-Earthquakes-2012-11-08/3wfw-mdbc/",  
  soc_query(  
    select = "region, avg(magnitude) as avg_magnitude, count(*) as count",  
    group_by = "region",  
    having = "count >= 5",  
    order_by = "avg_magnitude DESC"  
  )  
)
```

Index

`soc_discover`, [2](#)
`soc_metadata`, [4](#)
`soc_query`, [6](#)
`soc_read`, [7](#)