

Package: VancouvR (via r-universe)

September 24, 2026

Type Package

Title Access the 'City of Vancouver' Open Data API

Version 0.1.11

Maintainer Jens von Bergmann <jens@mountainmath.ca>

Description Wrapper around the 'City of Vancouver' Open Data API
<<https://opendata.vancouver.ca/api-console/explore/v2.1/>> to
simplify and standardize access to 'City of Vancouver' open
data. Functionality to list the data catalogue and access data
and geographic records.

License MIT + file LICENSE

Encoding UTF-8

NeedsCompilation no

Depends R (>= 4.2)

Imports dplyr, httr2 (>= 1.2.0), rlang, readr, digest, sf, geojsonsf,
tibble, purrr

Suggests knitr, rmarkdown, ggplot2, lwgeom, scales, tidyr, testthat
(>= 3.2.0), withr

VignetteBuilder knitr

URL <https://github.com/mountainMath/VancouvR>,
<https://mountainmath.github.io/VancouvR/>

BugReports <https://github.com/mountainMath/VancouvR/issues>

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-
dev libsqlite3-dev libudunits2-dev libx11-dev

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

Date/Publication 2026-08-25 00:46:35 UTC

RemoteUrl <https://github.com/mountainmath/vancouvr>

RemoteRef HEAD

RemoteSha e8ef5b1c6e243af7c4ad9e5513bd264f8ff6207f

Contents

aggregate_cov_data	2
get_cov_data	3
get_cov_facets	6
get_cov_metadata	7
get_cov_rate_limit	8
list_cov_datasets	9
list_cov_facets	10
search_cov_datasets	11

Index	13
--------------	-----------

aggregate_cov_data	<i>Aggregate data from the Vancouver Open Data Portal</i>
--------------------	---

Description

Sends a server-side aggregation query to the CoV Open Data API and returns the result as a tibble. Because aggregation is performed by the API, this is suitable for summarising large datasets without downloading all records.

Results are cached for the duration of the R session.

Grouped queries are answered by the dataset export endpoint and so are not subject to a record limit; every group is returned in a single request.

Usage

```
aggregate_cov_data(
  dataset_id,
  select = "count(*) as count",
  group_by = NULL,
  where = NULL,
  order_by = NULL,
  refine = NULL,
  exclude = NULL,
  limit = NULL,
  apikey = getOption("VancouverOpenDataApiKey"),
  refresh = FALSE
)
```

Arguments

dataset_id	Dataset id from the Vancouver Open Data catalogue
select	Aggregation expression using ODSQL syntax . Default <code>"count(*) as count"</code> .
group_by	Grouping expression using ODSQL syntax . Default <code>'NULL'</code> (no grouping).
where	Filter expression using ODSQL syntax . Default <code>'NULL'</code> (no filter).

order_by	Sort expression using ODSQL syntax , naming an aggregate from 'select' or a field from 'group_by', e.g. '"count DESC"'. Default 'NULL'.
refine	Facet filter(s) of the form '"field:value"'; see [get_cov_data()]. Default 'NULL'.
exclude	Facet exclusion(s) of the form '"field:value"'. Default 'NULL'.
limit	Maximum number of groups to return. Default 'NULL' returns all groups. Ignored when 'group_by' is 'NULL', which always yields one row.
apikey	Vancouver Open Data API key, default 'getOption("VancouverOpenDataApiKey")'
refresh	Bypass the session cache and re-download, default 'FALSE'

Value

A tibble with one row per group, with columns named according to the 'select' expression. Returns 'NULL' with a warning if the API cannot be reached.

See Also

[get_cov_data()] to download full or filtered records, [search_cov_datasets()] to find dataset IDs

Examples

```
# Count of each ticket status for fire hydrant infractions
aggregate_cov_data("parking-tickets-2017-2019",
  group_by = "status",
  where = "infractiontext LIKE 'FIRE'")

# Sum land and building values by tax year (server-side, no full download needed)
aggregate_cov_data("property-tax-report",
  select = "sum(current_land_value) as Land,
  sum(current_improvement_value) as Building",
  group_by = "tax_assessment_year")

# The ten most common tree genera
aggregate_cov_data("public-trees",
  group_by = "genus_name",
  order_by = "count DESC",
  limit = 10)
```

Description

Downloads a dataset and returns it as a tibble or 'sf' object. When 'cast_types = TRUE' (the default), field types are looked up via [get_cov_metadata()] and columns are automatically cast to integer, numeric, or Date.

Datasets whose metadata declares a 'geo_shape' or 'geo_point_2d' field are downloaded as **Flat-Geobuf** and returned as an 'sf' object with its coordinate reference system already set, rather than as CSV with the geometry re-parsed from text. The raw geometry fields are not part of that export, so such datasets come back with a single 'geometry' column and no 'geom' or 'geo_point_2d' column. Setting 'cast_types = FALSE' or 'use_labels = TRUE' downloads the CSV instead, and returns a plain tibble.

Results are cached for the duration of the R session, keyed on all query parameters. Re-running the same call does not trigger a second download.

Usage

```
get_cov_data(
  dataset_id,
  select = "*",
  where = NULL,
  order_by = NULL,
  refine = NULL,
  exclude = NULL,
  apikey = getOption("VancouverOpenDataApiKey"),
  rows = NULL,
  cast_types = TRUE,
  use_labels = FALSE,
  timezone = NULL,
  refresh = FALSE,
  ...
)
```

Arguments

dataset_id	Dataset id from the Vancouver Open Data catalogue
select	Column selection / expression string using ODSQL syntax , e.g. "current_land_value, land_coordinate as coord". Default "*" returns all columns.
where	Filter expression using ODSQL syntax , e.g. "tax_assessment_year='2024' AND zoning_district LIKE 'RS-'". Default 'NULL' returns all rows.
order_by	Sort expression using ODSQL syntax , e.g. "height_m DESC". Default 'NULL' leaves the portal's ordering.
refine	Facet filter(s) of the form "field:value", e.g. "genus_name:ACER". Values must match the facet exactly. Pass a character vector to apply several; multiple values on the same field are combined with OR, different fields with AND. Default 'NULL'.
exclude	Facet exclusion(s), in the same "field:value" form as 'refine'. Default 'NULL'.
apikey	Vancouver Open Data API key, default 'getOption("VancouverOpenDataApiKey")'

rows	Maximum number of rows to return. Default 'NULL' returns all rows.
cast_types	Logical; use metadata to auto-cast column types and to download spatial datasets as 'sf'. Default 'TRUE'.
use_labels	Logical; name the columns using the human-readable field labels instead of the API field names. Default 'FALSE'. Setting this to 'TRUE' disables type casting, as metadata is keyed on the API names.
timezone	Timezone used to render datetime fields, e.g. "America/Vancouver". Default 'NULL' uses the portal default (UTC).
refresh	Bypass the session cache and re-download, default 'FALSE'
...	Ignored; retained for compatibility with earlier versions

Value

A tibble, or an 'sf' object when the dataset has a spatial field and 'cast_types = TRUE'. Returns 'NULL' with a warning if the API cannot be reached.

See Also

[get_cov_metadata()] for field names and types, [aggregate_cov_data()] for server-side aggregation, [search_cov_datasets()] to find dataset IDs

Examples

```
# Select specific columns and limit rows (useful for exploration)
get_cov_data("property-tax-report",
  select = "tax_assessment_year, current_land_value, zoning_district",
  where = "tax_assessment_year = '2024'",
  rows = 10)

# The ten tallest maples, sorted server-side
get_cov_data("public-trees",
  refine = "genus_name:ACER",
  order_by = "height_m DESC",
  rows = 10)

# Spatial dataset: returned automatically as an sf object
property_polygons <- get_cov_data("property-parcel-polygons", rows = 10)
class(property_polygons) # "sf" "data.frame"

## Not run:
# Whole filtered datasets can be large, so they are not run here
get_cov_data("parking-tickets-2017-2019",
  where = "block = 1100 AND street = 'ALBERNI ST'")

## End(Not run)
```

get_cov_facets

Get the facet values of a CoV open data dataset

Description

Returns the values available for each faceted field, together with the number of records carrying them. This is a quick way to discover what a field actually contains before writing a ‘where’ clause or a ‘refine’ filter for [get_cov_data()].

The portal returns only the most common values of each facet (currently the top 100). Use [aggregate_cov_data()] with a ‘group_by’ for an exhaustive count.

Results are cached for the duration of the R session.

Usage

```
get_cov_facets(
  dataset_id,
  facet = NULL,
  where = NULL,
  refine = NULL,
  exclude = NULL,
  apikey = getOption("VancouverOpenDataApiKey"),
  refresh = FALSE
)
```

Arguments

dataset_id	the CoV open data dataset id
facet	Name(s) of the fields to facet on. Default ‘NULL’ returns every faceted field in the dataset.
where	Filter expression using ODSQL syntax , restricting the records the counts are computed over. Default ‘NULL’.
refine	Facet filter(s) of the form “field:value”; see [get_cov_data()]. Default ‘NULL’.
exclude	Facet exclusion(s) of the form “field:value”. Default ‘NULL’.
apikey	the CoV open data API key, optional
refresh	Bypass the session cache and re-download, default ‘FALSE’

Value

A tibble with columns ‘facet’ (the field name), ‘value’, and ‘count’. Returns ‘NULL’ with a warning if the API cannot be reached.

See Also

[get_cov_metadata()] for the list of fields, [list_cov_facets()] for the facets of the catalogue itself, [aggregate_cov_data()] for complete server-side counts

Examples

```
# What values does the genus field take?
get_cov_facets("public-trees", facet = "genus_name")

# Restricted to trees planted since 2020
get_cov_facets("public-trees", facet = "genus_name",
  where = "date_planted >= date('2020-01-01')")
```

get_cov_metadata	<i>Get field-level metadata for a CoV open data dataset</i>
------------------	---

Description

Returns a tibble describing each field in the dataset, including its API name, data type, display label, and description. Results are cached for the duration of the R session.

This function is called internally by [get_cov_data()] when ‘cast_types = TRUE’ to determine column types and identify spatial fields.

Usage

```
get_cov_metadata(
  dataset_id,
  apikey = getOption("VancouverOpenDataApiKey"),
  refresh = FALSE
)
```

Arguments

dataset_id	the CoV open data dataset id
apikey	the CoV open data API key, optional
refresh	Bypass the session cache and re-download, default ‘FALSE’

Value

A tibble with one row per field and columns:

name Field name as used in ‘where’ and ‘select’ queries

type API data type (e.g. “text”, “int”, “double”, “date”, “geo_shape”)

label Human-readable display label

description Field description, if provided by the portal

Returns ‘NULL’ with a warning if the API cannot be reached.

See Also

[get_cov_data()], [list_cov_datasets()]

Examples

```
# View all fields in the public trees dataset
get_cov_metadata("public-trees")

## Not run:
# Find which fields are spatial
get_cov_metadata("property-parcel-polygons") |>
  dplyr::filter(type == "geo_shape")

## End(Not run)
```

get_cov_rate_limit	<i>Report the remaining CoV open data API quota</i>
--------------------	---

Description

The City of Vancouver Open Data portal reports a daily request quota on every response. This returns the quota seen on the most recent request made by this package, so it reflects usage across every ‘Vancouver’ call in the session.

The quota is per API key, or per IP address when no key is set, and is shared with any other tool making requests under the same identity. ‘Vancouver’ warns once per session when fewer than 5

Usage

```
get_cov_rate_limit()
```

Value

A tibble with columns ‘limit’, ‘remaining’ and ‘reset’ (the time the quota resets, as reported by the portal), or ‘NULL’ if no request has been made yet in this session.

See Also

[get_cov_data()], [aggregate_cov_data()]

Examples

```
get_cov_data("public-trees", rows = 5)
get_cov_rate_limit()
```

list_cov_datasets	<i>List all datasets in the CoV open data catalogue</i>
-------------------	---

Description

Fetches the full City of Vancouver Open Data catalogue and returns it as a tibble. Results are cached for the duration of the R session; subsequent calls return the cached copy unless `refresh = TRUE`.

Usage

```
list_cov_datasets(
  trim = TRUE,
  where = NULL,
  refine = NULL,
  exclude = NULL,
  apikey = getOption("VancouverOpenDataApiKey"),
  refresh = FALSE
)
```

Arguments

trim	Remove columns that are entirely 'NA', default 'TRUE'
where	Filter expression applied by the portal, using ODSQL syntax over the catalogue's own fields, e.g. <code>"search(title, 'tree')"</code> or <code>"features LIKE 'geo'"</code> . Default 'NULL' returns the whole catalogue.
refine	Catalogue facet filter(s) of the form <code>"facet:value"</code> , e.g. <code>"theme:Sustainability"</code> . Values must match the facet exactly; use <code>[list_cov_facets()]</code> to discover them. Pass a character vector to apply several; multiple values on the same facet are combined with OR, different facets with AND. Default 'NULL'.
exclude	Catalogue facet exclusion(s), in the same <code>"facet:value"</code> form as 'refine'. Default 'NULL'.
apikey	the CoV open data API key, optional
refresh	Bypass the session cache and re-download, default 'FALSE'

Value

A tibble with one row per dataset. The first four columns are always `'dataset_id'`, `'title'`, `'keyword'`, and `'search-term'`; remaining columns contain catalogue metadata (trimmed to non-empty columns when `trim = TRUE`). Returns 'NULL' with a warning if the API cannot be reached.

See Also

`[list_cov_facets()]` to discover the values 'refine' accepts, `[search_cov_datasets()]` to filter the catalogue by a search term, `[get_cov_data()]` to download a specific dataset

Examples

```
# Only the datasets that carry geographic records
list_cov_datasets(where = "features LIKE 'geo'")

# Datasets filed under a catalogue theme
list_cov_datasets(refine = "theme:Sustainability")

## Not run:
# The entire catalogue
list_cov_datasets()

## End(Not run)
```

list_cov_facets	<i>List the facets of the CoV open data catalogue</i>
-----------------	---

Description

Returns the values the catalogue can be filtered on, together with the number of datasets carrying each. This is the starting point for browsing the portal rather than searching it: it answers what themes and keywords exist before passing them to [list_cov_datasets()] as a ‘refine’ filter.

The catalogue facets are ‘theme’, ‘keyword’, ‘features’, ‘custom.data-owner’ and ‘custom.data-team’. The ‘features’ facet is worth knowing about: its ‘geo’ value identifies the datasets [get_cov_data()] returns as ‘sf’ objects, and ‘timeserie’ those carrying a date field.

Results are cached for the duration of the R session.

Usage

```
list_cov_facets(
  facet = NULL,
  where = NULL,
  refine = NULL,
  exclude = NULL,
  apikey = getOption("VancouverOpenDataApiKey"),
  refresh = FALSE
)
```

Arguments

facet	Name(s) of the catalogue facets to return. Default ‘NULL’ returns every facet.
where	Filter expression using ODSQL syntax , restricting the datasets the counts are computed over. Default ‘NULL’.
refine	Facet filter(s) of the form “facet:value”. Default ‘NULL’.
exclude	Facet exclusion(s) of the form “facet:value”. Default ‘NULL’.
apikey	the CoV open data API key, optional
refresh	Bypass the session cache and re-download, default ‘FALSE’

Value

A tibble with columns ‘facet’ (the facet name), ‘value’, and ‘count’ (the number of datasets). Returns ‘NULL’ with a warning if the API cannot be reached.

See Also

[list_cov_datasets()] to filter the catalogue on these values, [get_cov_facets()] for the facets of an individual dataset

Examples

```
# Every facet of the catalogue
list_cov_facets()

# Just the themes, and how many datasets each holds
list_cov_facets(facet = "theme")
```

search_cov_datasets	<i>Search the CoV open data catalogue</i>
---------------------	---

Description

Filters the City of Vancouver Open Data catalogue for datasets whose title, dataset ID, keyword, or search-term fields match ‘search_term’ (using ‘grepl()’, so regular expressions are supported). When no exact match is found, a fuzzy-match hint list of similarly named datasets is printed.

Usage

```
search_cov_datasets(
  search_term,
  trim = TRUE,
  apikey = getOption("VancouverOpenDataApiKey"),
  refresh = FALSE
)
```

Arguments

search_term	A grep-compatible string to search through dataset titles, IDs, keywords, and search terms
trim	Remove columns that are entirely ‘NA’, default ‘TRUE’
apikey	the CoV open data API key, optional
refresh	Bypass the session cache and re-download, default ‘FALSE’

Value

A tibble with one row per matching dataset, in the same format as `[list_cov_datasets()]`. Returns 'NULL' with a warning if the API cannot be reached.

See Also

`[list_cov_datasets()]` to retrieve the full catalogue, `[get_cov_data()]` to download a specific dataset

Examples

```
# Search using a plain string
search_cov_datasets("trees")

# Search using a regular expression
search_cov_datasets("parking.*(2019|2020)")
```

Index

`aggregate_cov_data`, [2](#)
`get_cov_data`, [3](#)
`get_cov_facets`, [6](#)
`get_cov_metadata`, [7](#)
`get_cov_rate_limit`, [8](#)
`list_cov_datasets`, [9](#)
`list_cov_facets`, [10](#)
`search_cov_datasets`, [11](#)