

# Package ‘VancouvR’

May 7, 2026

**Type** Package

**Title** Access the 'City of Vancouver' Open Data API

**Version** 0.1.9

**Maintainer** Jens von Bergmann <jens@mountainmath.ca>

**Description**

Wrapper around the 'City of Vancouver' Open Data API <<https://opendata.vancouver.ca/api/v2/console>> 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

**RoxygenNote** 7.3.3

**Depends** R (>= 4.2)

**Imports** dplyr, httr, rlang, urltools, readr, digest, sf, geojsonsf, tibble, purrr

**Suggests** knitr, rmarkdown, ggplot2, lwgeom, scales, tidyr, testthat (>= 3.0.0)

**VignetteBuilder** knitr

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

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

**Author** Jens von Bergmann [aut, cre]

**Repository** CRAN

**Date/Publication** 2026-03-04 03:30:11 UTC

## Contents

aggregate_cov_data . . . . .	2
get_cov_data . . . . .	3

get_cov_metadata . . . . .	4
list_cov_datasets . . . . .	5
search_cov_datasets . . . . .	6

<b>Index</b>	<b>8</b>
--------------	----------

---

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.

## Usage

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

## Arguments

dataset_id	Dataset id from the Vancouver Open Data catalogue
select	Aggregation expression using <b>ODSQL syntax</b> . Default <code>"count(*) as count"</code> .
group_by	Grouping expression using <b>ODSQL syntax</b> . Default <code>'NULL'</code> (no grouping).
where	Filter expression using <b>ODSQL syntax</b> . Default <code>'NULL'</code> (no filter).
apikey	Vancouver Open Data API key, default <code>'getOption("VancouverOpenDataApiKey")'</code>
refresh	Bypass the session cache and re-download, default <code>'FALSE'</code>

## Value

A tibble with one row per group, with columns named according to the `'select'` expression.

## See Also

[get\_cov\_data()] to download full or filtered records, [search\_cov\_datasets()] to find dataset IDs

**Examples**

```
## Not run:
# Count of each ticket status for fire hydrant infractions
aggregate_cov_data("parking-tickets-2017-2019",
  group_by = "status",
  where = "infraciontext 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")

## End(Not run)
```

---

get\_cov\_data

---

*Download a dataset from the Vancouver Open Data Portal*


---

**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 containing a 'geo\_shape' field are returned as an 'sf' object; if spatial conversion fails a plain tibble is returned with a warning.

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,
  apikey = getOption("VancouverOpenDataApiKey"),
  rows = NULL,
  cast_types = TRUE,
  refresh = FALSE,
  ...
)
```

**Arguments**

dataset_id	Dataset id from the Vancouver Open Data catalogue
select	Column selection / expression string using <b>ODSQL syntax</b> , e.g. "current_land_value, land_coordinate as coord". Default "*" returns all columns.

where	Filter expression using <b>ODSQL syntax</b> , e.g. <code>"tax_assessment_year='2024' AND zoning_district LIKE 'RS-'"</code> . Default <code>'NULL'</code> returns all rows.
apikey	Vancouver Open Data API key, default <code>'getOption("VancouverOpenDataApiKey")'</code>
rows	Maximum number of rows to return. Default <code>'NULL'</code> returns all rows.
cast_types	Logical; use metadata to auto-cast column types and convert spatial datasets to <code>'sf'</code> . Default <code>'TRUE'</code> .
refresh	Bypass the session cache and re-download, default <code>'FALSE'</code>
...	Ignored; retained for compatibility with earlier versions

**Value**

A tibble, or an `'sf'` object when the dataset has a `'geo_shape'` field and `'cast_types = TRUE'`.

**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**

```
## Not run:
# Filtered download: parking tickets on one block
get_cov_data("parking-tickets-2017-2019",
             where = "block = 1100 AND street = 'ALBERNI ST'")

# 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 = 100)

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

## End(Not run)
```

---

get\_cov\_metadata

*Get field-level metadata for a CoV open data dataset*

---

**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

**See Also**

[get\_cov\_data()], [list\_cov\_datasets()]

**Examples**

```
## Not run:  
# View all fields in the street trees dataset  
get_cov_metadata("street-trees")  
  
# Find which fields are spatial  
get_cov_metadata("property-parcel-polygons") |>  
  dplyr::filter(type == "geo_shape")  
  
## End(Not run)
```

---

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

**Arguments**

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 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').

**See Also**

[search\_cov\_datasets()] to filter the catalogue by a search term, [get\_cov\_data()] to download a specific dataset

**Examples**

```
## Not run:
list_cov_datasets()

## End(Not run)
```

---

search\_cov\_datasets    *Search the CoV open data catalogue*

---

**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()]`.

**See Also**

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

**Examples**

```
## Not run:  
# Search using a plain string  
search_cov_datasets("trees")  
  
# Search using a regular expression  
search_cov_datasets("parking.*(2019|2020)")  
  
## End(Not run)
```

# Index

`aggregate_cov_data`, [2](#)

`get_cov_data`, [3](#)

`get_cov_metadata`, [4](#)

`list_cov_datasets`, [5](#)

`search_cov_datasets`, [6](#)