

# Package: VancouvR (via r-universe)

September 17, 2024

**Type** Package

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

**Version** 0.1.8

**Author** Jens von Bergmann

**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** 7.2.3

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

**Suggests** knitr, rmarkdown, ggplot2, lwgeom, tidyr

**VignetteBuilder** knitr

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

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

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

**RemoteUrl** <https://github.com/mountainmath/vancouvR>

**RemoteRef** HEAD

**RemoteSha** 566f542325da3000ff94d6eb36ade7e74720a59c

## Contents

aggregate_cov_data . . . . .	2
get_cov_data . . . . .	3
get_cov_metadata . . . . .	4
list_cov_datasets . . . . .	5
search_cov_datasets . . . . .	5
<b>Index</b>	<b>7</b>

---

aggregate_cov_data	<i>Get aggregates from dataset from Vancouver Open Data Portal</i>
--------------------	--

---

## Description

Get aggregates from dataset from Vancouver Open Data Portal

## 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	select string for aggregation, default is 'count(*) as count' It accepts <b>ODSQL syntax</b> .
group_by	grouping variables for the query It accepts <b>ODSQL syntax</b> .
where	Query parameter to filter data (default 'NULL' no filter) It accepts <b>ODSQL syntax</b> .
apikey	Vancouver Open Data API key, default 'getOption("VancouverOpenDataApiKey")'
refresh	refresh cached data, default 'FALSE'

## Value

tibble format data table output

**Examples**

```
# Count all parking tickets that relate to fire hydrants by ticket status
## Not run:
aggregate_cov_data("parking-tickets-2017-2019",
                  group_by = "status",
                  where = "infraciontext LIKE 'FIRE'")

## End(Not run)
```

---

get\_cov\_data

*Get datasets from Vancouver Open Data Portal*


---

**Description**

Get datasets from Vancouver Open Data Portal

**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	select string for fields to return, returns all fields by default. It accepts <b>ODSQL syntax</b> .
where	Query parameter to filter data (default 'NULL' no filter) It accepts <b>ODSQL syntax</b> .
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 look up types and type-cast automatically, default 'TRUE'
refresh	refresh cached data, default 'FALSE'
...	optional ignored parameters, for compatibility with previous versions that relied on the 'format' parameter

**Value**

tibble or sf object data table output, depending on whether the dataset is spatial and 'cast\_types' is 'TRUE'

**Examples**

```
# Get all parking tickets issued at the 1100 block of Alberni Street between 2017 and 2019
## Not run:
get_cov_data("parking-tickets-2017-2019", where = "block = 1100 AND street = 'ALBERNI ST'")

## End(Not run)
```

---

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

---

**Description**

Get metadata for CoV open data dataset

**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	refresh cached data, default 'FALSE'

**Value**

tibble format data table output

**Examples**

```
# Get the metadata for the street trees dataset
## Not run:
get_cov_metadata("street-trees")

## End(Not run)
```

---

list\_cov\_datasets      *Download the CoV open data catalogue*

---

**Description**

Download the CoV open data catalogue

**Usage**

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

**Arguments**

trim	trim all NA columns, optional, default 'TRUE'
apikey	the CoV open data API key, optional
refresh	refresh cached data, default 'FALSE'

**Value**

tibble format data table output

**Examples**

```
# List and search available datasets  
## Not run:  
list_cov_datasets()  
  
## End(Not run)
```

---

search\_cov\_datasets      *Search for CoV open data datasets*

---

**Description**

Search for CoV open data datasets

**Usage**

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

**Arguments**

search_term	grep string to search through datasets
trim	trim all NA columns, optional, default 'TRUE'
apikey	the CoV open data API key, optional
refresh	refresh cached data, default 'FALSE'

**Value**

tibble format data table output

**Examples**

```
# search available datasets relating to trees  
## Not run:  
search_cov_datasets("trees")  
  
## End(Not run)
```

# Index

`aggregate_cov_data`, [2](#)

`get_cov_data`, [3](#)

`get_cov_metadata`, [4](#)

`list_cov_datasets`, [5](#)

`search_cov_datasets`, [5](#)