Package 'opendatatoronto'

March 29, 2020

Title Access the City of Toronto Open Data Portal

Version 0.1.3

Description Access data from the ``City of Toronto
Open Data Portal" (<https://open.toronto.ca>) directly from R.

License MIT + file LICENSE

URL https://sharlagelfand.github.io/opendatatoronto/,
https://github.com/sharlagelfand/opendatatoronto/

BugReports https://github.com/sharlagelfand/opendatatoronto/issues

Depends R (>= 2.10)

Imports ckanr (>= 0.4.0), magrittr, readxl, sf, tibble, xml2, curl

Suggests covr, dplyr, ggplot2, knitr, purrr, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Sharla Gelfand [aut, cre],
City of Toronto [cph, fnd]

Maintainer Sharla Gelfand <sharla.gelfand@gmail.com>

Repository CRAN

Date/Publication 2020-03-29 18:30:02 UTC

R topics documented:

`opendatatoronto-package`
opendatatoronto

Description

opendatatoronto is an R interface to the City of Toronto's Open Data Portal. The goal of the package is to help read data directly into R without needing to manually download it via the portal.

Details

For more information, please start with the vignettes: `browseVignettes(package = "opendatatoronto")` or visit the package website.

Author(s)

Maintainer: Sharla Gelfand <sharla.gelfand@gmail.com>

Other contributors:

- City of Toronto [copyright holder, funder]

See Also

Useful links:

- https://sharlagelfand.github.io/opendatatoronto/
- https://github.com/sharlagelfand/opendatatoronto/
- Report bugs at https://github.com/sharlagelfand/opendatatoronto/issues

browse_package *Open the package's page in your browser*

Description

Opens a browser to the package's page on the City of Toronto Open Data Portal.

Usage

```
browse_package(package)
```

Arguments

package	A way to identify the package. Either a package ID (passed as a character vector directly), a single package resulting from `list_packages` or `search_packages`, or the package's URL from the portal.

Value

A browser is opened to the package's page on the City of Toronto Open Data Portal website if the session is interactive. The URL is returned as a character string.

Examples

```
ttc_subway_delays <- search_packages("ttc subway delay")
browse_package(ttc_subway_delays)
browse_package("https://open.toronto.ca/dataset/business-improvement-areas/")
```

browse_portal *Open the City of Toronto Open Data Portal in your browser*

Description

Opens a browser to `https://open.toronto.ca`.

Usage

```
browse_portal()
```

Value

A browser is opened to the City of Toronto Open Data Portal website if the session is interactive. The URL is returned as a character string.

Examples

```
browse_portal()
```

`browse_resource` *Open the resource's package page in your browser*

Description

Opens a browser to the resource's package page on the City of Toronto Open Data Portal.

Usage

```
browse_resource(resource)
```

Arguments

`resource`	A way to identify the resource. Either a resource ID (passed as a character vector directly) or a single resource resulting from `list_package_resources`.

Value

A browser is opened to the resource's package page on the City of Toronto Open Data Portal website if the session is interactive. The URL is returned as a character string.

Examples

```
ttc_subway_delays <- search_packages("ttc subway delay")
res <- list_package_resources(ttc_subway_delays)
browse_resource(res[1, ])
```

`get_resource` *Download a resource into your R session*

Description

Download a resource from the portal directly into your R session. CSV, XLS, XLSX, XML, JSON, SHP, ZIP, and GeoJSON resources are supported.

Usage

```
get_resource(resource)
```

Arguments

resource	A way to identify the resource. Either a resource ID (passed as a character vector directly) or a single resource resulting from list_package_resources.

Value

In most cases, the resource is returned as a tibble or list of tibbles. If it is a spatial resource (i.e. SHP or GeoJSON), it is returned as an sf object.

Examples

```
list_package_resources("https://open.toronto.ca/dataset/neighbourhoods/") %>%
get_resource()
```

---

list_packages *List packages*

Description

List packages available on the portal.

Usage

```
list_packages(limit = 50)
```

Arguments

limit	The maximum number of packages to return. The default is 50.

Value

A tibble of available packages and metadata, including title, id, topics, civic_issues, dataset_category, num_resources (the number of resources in the package), formats (the different formats of the resources), refresh_rate (how often the package is refreshed), and last_refreshed (the date it was last refreshed).

Examples

```
list_packages(5)
```

list_package_resources *List resources for a package*

Description

List resources for a package on the portal.

Usage

```
list_package_resources(package)
```

Arguments

package	A way to identify the package. Either a package ID (passed as a character vector directly), a single package resulting from list_packages or search_packages, or the package's URL from the portal.

Value

A tibble of resources along with metadata, including name, id, format (the format of the resource file), and last_modified (the date the resource was last modified).

Examples

```
list_package_resources("1db34737-ffad-489d-a590-9171d500d453")
list_package_resources("https://open.toronto.ca/dataset/ttc-subway-delay-data")
```

search_packages *Search packages by title*

Description

Search portal packages by title.

Usage

```
search_packages(title, limit = 50)
```

Arguments

title	Title to search (case-insensitive).
limit	Maximum number of packages to return. The default is 50. The maximum limit is 1000.

Value

A tibble of matching packages along with package metadata, including `title`, `id`, `topics`, `civic_issues`, `dataset_category`, `num_resources` (the number of resources in the package), `formats` (the different formats of the resources), `refresh_rate` (how often the package is refreshed), and `last_refreshed` (the date it was last refreshed).

Examples

```
search_packages("ttc")
```

---

show_package *Show a package's metadata*

Description

Show a portal package's metadata.

Usage

```
show_package(package)
```

Arguments

package	A way to identify the package. Either a package ID (passed as a character vector directly) or the package's URL from the portal.

Value

A tibble including `title`, `id`, `topics`, `civic_issues`, `dataset_category`, `num_resources` (the number of resources in the package), `formats` (the different formats of the resources), `refresh_rate` (how often the package is refreshed), and `last_refreshed` (the date it was last refreshed).

Examples

```
show_package("c01c6d71-de1f-493d-91ba-364ce64884ac")
```

Index