Package ‘threesixtygiving’

December 2, 2020

**Title**  Download Charitable Grants from the ‘360Giving’ Platform

**Version**  0.2.2

**Description**  Access open data from <https://www.threesixtygiving.org>, a database of charitable grant giving in the UK operated by ’360Giving’. The package provides functions to search and retrieve data on charitable grant giving, and process that data into tidy formats. It relies on the ’360Giving’ data standard, described at <https://standard.threesixtygiving.org/>.

**URL**  https://docs.evanodell.com/threesixtygiving,  
https://github.com/evanodell/threesixtygiving,

**BugReports**  https://github.com/evanodell/threesixtygiving/issues

**License**  GPL-3

**Encoding**  UTF-8

**LazyData**  true

**Imports**  dplyr, httr, readxl, readr, jsonlite, purrr, janitor, tidyr, tidyselect, anytime, tibble, curl

**RoxygenNote**  7.1.1

**Suggests**  covr, testthat (>= 2.1.0), knitr, rmarkdown, ggplot2, scales, stringi, ggrepel, fixerapi

**VignetteBuilder**  knitr

**NeedsCompilation**  no

**Author**  Evan Odell [aut, cre] (<https://orcid.org/0000-0003-1845-808X>)

**Maintainer**  Evan Odell <evanodell191@gmail.com>

**Repository**  CRAN

**Date/Publication**  2020-12-02 09:30:02 UTC
threesixtygiving package

Description

360Giving is a data standard for publishing information about charitable grant giving in the UK. threesixtygiving provides functions to retrieve and process charitable grant funding data from 360Giving.

Details

See the 360 Giving website for more information on the source of this data, and the 360Giving Standard page for details of the data standard this package relies on.

tsg_all_grants

Description

Returns a list of data frames with details of all grants from funders returned by tsg_available().

Usage

tsg_all_grants(verbos = TRUE, timeout = 30, retries = 0, correct_names = TRUE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>verbose</td>
<td>If TRUE, prints console messages on data retrieval progress. Defaults to TRUE.</td>
</tr>
<tr>
<td>timeout</td>
<td>The maximum request time, in seconds. If data is not returned in this time, a value of NA is returned for that dataset. Defaults to 30 seconds.</td>
</tr>
<tr>
<td>retries</td>
<td>The number of retries to make if a request is not successful. Defaults to 0.</td>
</tr>
<tr>
<td>correct_names</td>
<td>If TRUE, corrects known mistakes in column names, such as spelling mistakes. Defaults to TRUE.</td>
</tr>
</tbody>
</table>
tsg_available

Details

Due to the structure of the 360 Giving data, the package will make multiple attempts to request data. This can cause issues with servers automatically blocking requests. You can use `tsg_missing()` to identify missing sets of grant data.

Value

A list of tibbles with grant data.

Examples

```r
## Not run:
all_grants <- tsg_all_grants()
## End(Not run)
```

---

tsg_available

Available datasets

Description

Returns a tibble with information on all datasets available through 360Giving. Some funders have multiple datasets.

Usage

`tsg_available()`

Value

A tibble with details on all available datasets.

Examples

```r
## Not run:
available <- tsg_available()
## End(Not run)
```
**tsg_core_data**  
*Core Data*

**Description**

Given a list returned by `tsg_all_grants()` or `tsg_search_grants()`, creates a tibble with the core variables required by the 360Giving open standard, as well as the publisher prefix and dataset identifier, which are useful for data processing, and the licence the data was published under.

**Usage**

```
tsg_core_data(x, verbose = TRUE)
```

**Arguments**

- `x`  
  A list of tibble with grant data returned by `tsg_all_grants()`.
- `verbose`  
  If TRUE, prints console messages on data retrieval progress. Defaults to TRUE.

**Value**

A tibble with the core variables in the 360Giving standard.

**See Also**

- `tsg_process_data()`, which does the same processing but returns all available variables.

**Examples**

```r
## Not run:
grants <- tsg_all_grants()
df <- tsg_core_data(grants)
## End(Not run)
```

---

**tsg_missing**  
*Identify missing data sets*

**Description**

Returns a tibble with details on any available data missing from a list of grants returned by `tsg_all_grants()`.

**Usage**

```
tsg_missing(x)
```
tsg_process_data

Arguments

x
A list of tibble with grant data returned by tsg_all_grants().

Value

A tibble with details on funders missing data from a list of tibbles passed to the function.

Examples

## Not run:
all_grants <- tsg_all_grants()
missing_grants <- tsg_missing(all_grants)
## End(Not run)

---

### tsg_process_data

Convert data to tibble

Description

Given a list returned by tsg_all_grants() or tsg_search_grants(), creates a tibble with all available variables. This tibble contains roughly 200 columns if all available grants are used (as of October 2019), due to differences in how grant data is labelled and structured.

Usage

tsg_process_data(x, min_coverage = 0, verbose = TRUE)

Arguments

x
A list of tibble with grant data returned by tsg_all_grants().

min_coverage
A number from 0 to 1. If >0, only returns variables with a value other than NA for at least that proportion of rows. Defaults to 0 and returns all columns.

verbose
If TRUE, prints console messages on data retrieval progress. Defaults to TRUE.

Value

A tibble with all variables from all provided grants.

See Also

tsg_core_data(), which does the same processing but only returns the core variables in the 360Giving standard.
### tsg_search_funders

Search funders

**Description**

Return a tibble with information on all grant datasets where funder data matches one or more search strings.

**Usage**

```r
library(tsg)

# Search funders
result <- tsg_search_funders(
  search = c("example", "another example"),
  search_in = "funder_name",
  verbose = TRUE,
  ignore_case = TRUE,
  perl = FALSE,
  fixed = FALSE
)
```

**Arguments**

- **search**
  - The string(s) to search for. By default allows POSIX 1003.2 regular expressions. Use `perl = TRUE` for perl-style regex, or `fixed = TRUE` for fixed strings. Accepts single strings or a character vector of strings.

- **search_in**
  - The name of the column to search in. Accepts single strings or a character vector of column names. If NULL, searches all columns.

- **verbose**
  - If TRUE, prints console messages on data retrieval progress. Defaults to TRUE.

- **ignore_case**
  - If TRUE ignores case.

- **perl**
  - If TRUE, uses perl-style regex.

- **fixed**
  - If TRUE, searches will be matched as-is.

**Value**

A tibble with information on matching datasets

**Examples**

```r
## Not run:
grants <- tsg_all_grants()
df1 <- tsg_process_data(grants)
# Only return data from columns with more than 50% coverage
df2 <- tsg_process_data(grants, min_coverage = 0.5)
## End(Not run)
```
tsg_search_grants

See Also

`tsg_search_grants()` for retrieving all grants from matching funders.

Examples

```r
## Not run:
search1 <- tsg_search_funders(search = c("bbc", "caBinet"))
## End(Not run)
```

---

**tsg_search_grants**  
*Search for specific datasets*

Description

Returns a list of tibbles with details of all grants from specific funders.

Usage

```r
tsg_search_grants(
  search,
  search_in = NULL,
  verbose = TRUE,
  ignore_case = TRUE,
  perl = FALSE,
  fixed = FALSE,
  ...
)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>search</td>
<td>The string(s) to search for. By default allows POSIX 1003.2 regular expressions. Use perl = TRUE for perl-style regex, or fixed = TRUE for fixed strings. Accepts single strings or a character vector of strings.</td>
</tr>
<tr>
<td>search_in</td>
<td>The name of the column to search in. Accepts single strings or a character vector of column names. If NULL, searches all columns.</td>
</tr>
<tr>
<td>verbose</td>
<td>If TRUE, prints console messages on data retrieval progress. Defaults to TRUE.</td>
</tr>
<tr>
<td>ignore_case</td>
<td>If TRUE ignores case.</td>
</tr>
<tr>
<td>perl</td>
<td>If TRUE, uses perl-style regex.</td>
</tr>
<tr>
<td>fixed</td>
<td>If TRUE, searches will be matched as-is.</td>
</tr>
<tr>
<td>...</td>
<td>Additional params passed to <code>tsg_all_grants()</code></td>
</tr>
</tbody>
</table>
Details

tsg_search_grants retrieves grants where funder data matches one or more search strings. If only one dataset matches queries, returns a tibble of that dataset. Use tsg_specific_df() to pass a dataframe.

Value

A single tibble (if only one grant maker matches the queries) or a list of tibbles (if the query matches multiple datasets).

See Also

tsg_search_funders() for retrieving information on available datasets from matching funders.
tsg_specific_df() to retrieve data contained with all or specific rows of a tibble returned by tsg_available() or tsg_missing().

Examples

## Not run:
specific1 <- tsg_search_grants(search = c("bbc", "caBinet"))

## End(Not run)

---

tsg_specific_df | Get specific datasets

Description

Retrieve data contained with all or specific rows of a tibble returned by tsg_available() or tsg_missing().

Usage

tsg_specific_df(x, verbose = TRUE, timeout = 30, retries = 0)

Arguments

- **x**: All or a subset of a tibble returned by tsg_available() or tsg_missing()
- **verbose**: If TRUE, prints console messages on data retrieval progress. Defaults to TRUE.
- **timeout**: The maximum request time, in seconds. If data is not returned in this time, a value of NA is returned for that dataset. Defaults to 30 seconds.
- **retries**: The number of retries to make if a request is not successful. Defaults to 0.

Value

A list of tibbles with grant data.
tsg_specific_df

See Also

  tsg_search_grants()
  tsg_available()
  tsg_missing()

Examples

  ## Not run:
  all_grants <- tsg_all_grants()

  missing_grants <- tsg_missing(all_grants)

  more_grants <- tsg_specific_df(missing_grants)

  ## End(Not run)
Index

threesixtygiving, 2
threesixtygiving-package
  (threesixtygiving), 2

tsg_all_grants, 2
tsg_all_grants(), 4, 5, 7

tsg_available, 3
tsg_available(), 2, 8, 9

tsg_core_data, 4
tsg_core_data(), 5

tsg_missing, 4
tsg_missing(), 3, 8, 9

tsg_process_data, 5
tsg_process_data(), 4

tsg_search_funders, 6
tsg_search_funders(), 8

tsg_search_grants, 7
tsg_search_grants(), 4, 5, 7, 9

tsg_specific_df, 8
tsg_specific_df(), 8