Package 'statswalesr'

July 12, 2026

Type Package

Title Easily Extract Data from 'StatsWales'

Version 1.0.0

Description Download data from the 'StatsWales' public API (<https://api.stats.gov.wales/v2>) into R. Provides functions to list, search, and browse datasets by topic, retrieve data with optional filtering, sorting, and pivoting, and download datasets as CSV or Excel files.

License MIT + file LICENSE

URL https://github.com/jamie-ralph/statswalesr, https://jamie-ralph.github.io/statswalesr/

BugReports https://github.com/jamie-ralph/statswalesr/issues

Encoding UTF-8

Depends R (>= 4.1.0)

Imports httr2, jsonlite

Suggests testthat (>= 3.0.0)

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Jamie Ralph [aut, cre]

Maintainer Jamie Ralph <jamesryanralph@outlook.com>

Repository CRAN

Date/Publication 2026-07-12 14:40:02 UTC

Contents

statswales_create_query 2
statswales_download_dataset 3
statswales_get_dataset 4
statswales_get_filters 6

statswales_get_metadata 7
statswales_get_pivot 8
statswales_get_query 9
statswales_get_topic 10
statswales_list_datasets 11
statswales_list_topics 12
statswales_search 13

Index **14**

`statswales_create_query` *Create a stored query and return its filter ID*

Description

Submits a set of filters and display options to the API and returns the resulting filter ID. The same inputs always produce the same ID, so this is useful for building shareable, reproducible data requests.

Usage

```
statswales_create_query(
  dataset_id,
  filter = NULL,
  options = list(use_raw_column_names = FALSE, use_reference_values = FALSE,
    data_value_type = "formatted")
)
```

Arguments

`dataset_id`	A dataset UUID string. Use `statswales_list_datasets()` to find dataset IDs.
`filter`	Filter criteria in the same format as `statswales_get_dataset()`.
`options`	Display options in the same format as `statswales_get_dataset()`.

Details

Pass the returned `filter_id` to `statswales_get_query()` to inspect the query configuration, or use it directly in `statswales_get_dataset()` / `statswales_download_dataset()` workflows.

Value

A character string containing the 12-character filter ID, or `NULL` if the request fails.

Examples

```
## Not run:
datasets <- statswales_list_datasets()
id <- datasets$id[1]

fid <- statswales_create_query(id)
statswales_get_query(id, fid)

## End(Not run)
```

`statswales_download_dataset` *Download a dataset file*

Description

Downloads a published dataset as a file. The `filter` and `options` arguments work identically to `statswales_get_dataset()`.

Usage

```
statswales_download_dataset(
  dataset_id,
  format = "csv",
  lang = "en-gb",
  path = NULL,
  filter = NULL,
  options = list(use_raw_column_names = FALSE, use_reference_values = FALSE,
    data_value_type = "formatted")
)
```

Arguments

`dataset_id`	A dataset UUID string. Use `statswales_list_datasets()` to find dataset IDs.
`format`	File format. One of `"csv"` (default) or `"xlsx"`.
`lang`	Language for the downloaded file. One of `"en-gb"` (default), `"en"`, `"cy-gb"`, or `"cy"`.
`path`	File path for saving the download. If `NULL` (default), a temporary file is created automatically.
`filter`	Filter criteria in the same format as `statswales_get_dataset()`.
`options`	Display options in the same format as `statswales_get_dataset()`.

Value

The path to the downloaded file, invisibly. Returns `NULL` if the download fails.

Examples

```
## Not run:
datasets <- statswales_list_datasets()
id <- datasets$id[1]

# Download full dataset as CSV
path <- statswales_download_dataset(id)
df <- read.csv(path)

# Download filtered data as Excel
statswales_download_dataset(
  id,
  format = "xlsx",
  filter = list(list(Year = c("2022", "2023"))),
  path = "my_data.xlsx"
)

## End(Not run)
```

`statswales_get_dataset` *Retrieve data for a dataset*

Description

Returns a data frame containing rows from a published dataset. Filters and display options are submitted to the API as a stored query; the same inputs always produce the same query ID so repeated calls are efficient.

Usage

```
statswales_get_dataset(
  dataset_id,
  lang = "en-gb",
  page_number = 1,
  page_size = 10000,
  filter = NULL,
  options = list(use_raw_column_names = FALSE, use_reference_values = FALSE,
    data_value_type = "formatted"),
  sort_by = NULL,
  all_pages = TRUE,
  tidy = TRUE
)
```

Arguments

dataset_id	A dataset UUID string. Use statswales_list_datasets() to find dataset IDs.
lang	Language for text values. One of "en-gb" (default), "en", "cy-gb", or "cy".
page_number	Page of results to return. Default 1. Only used when all_pages = FALSE.
page_size	Rows per page. Default 10000 (the API maximum).
filter	A list of filter objects. Each element is a named list mapping a column name (from statswales_get_filters()) to a character vector of reference codes. Multiple list elements use AND logic; multiple codes within one element use OR logic. Example: list(list(Year = c("2020", "2021")), list(Area = c("W92000004"))).
options	A named list of display options: use_raw_column_names FALSE (default) returns human-readable column names; TRUE returns internal fact-table names. use_reference_values FALSE (default) returns human-readable values; TRUE returns reference codes. data_value_type One of "raw", "raw_extended", "formatted" (default), "formatted_extended", or "with_note_codes". Note: as of July 2026 the StatsWales API ignores use_raw_column_names and use_reference_values, so output always uses human-readable column names and values regardless of these settings. Only data_value_type currently changes the output.
sort_by	Optional sort order. Either a ready-made string in the API's "column:direction" format (e.g. "Year:desc"), or a named character vector such as c(Year = "desc", Area = "asc"). Directions are "asc" or "desc".
all_pages	If TRUE (default), automatically fetches and row-binds all pages so the entire dataset is returned. Set to FALSE to retrieve a single page controlled by page_number and page_size.
tidy	If TRUE (default), the result is cleaned for analysis: the API's internal *_sort columns are dropped, whitespace padding is stripped, and numeric-looking columns (such as the data values) are converted to numeric. Set to FALSE to return the API response as-is.

Value

A data frame of dataset rows, or NULL if the request fails.

Examples

```
## Not run:
datasets <- statswales_list_datasets()
id <- datasets$id[1]

# Entire dataset, human-readable (default)
df <- statswales_get_dataset(id)

# A single page of 100 rows
```

```
df_page <- statswales_get_dataset(id, all_pages = FALSE, page_size = 100)

# Filtered to specific years
filters <- statswales_get_filters(id)
df_filtered <- statswales_get_dataset(
  id,
  filter = list(list(Year = c("2020", "2021")))
)

## End(Not run)
```

`statswales_get_filters` *Get available filters for a dataset*

Description

Returns a list of filterable dimensions and their allowed values. Use the output to build the `filter` argument for `statswales_get_dataset()`, `statswales_download_dataset()`, and `statswales_get_pivot()`.

Usage

```
statswales_get_filters(dataset_id, lang = "en-gb")
```

Arguments

`dataset_id`	A dataset UUID string. Use `statswales_list_datasets()` to find dataset IDs.
`lang`	Language for returned text. One of `"en-gb"` (default), `"en"`, `"cy-gb"`, or `"cy"`.

Value

A list where each element corresponds to a filterable dimension and contains:

`factTableColumn` Internal column name used in data queries.

`columnName` Human-readable dimension name — use this as the key in filter objects.

`values` A data frame of available filter values with columns `reference`, `description`, `parent`, and `level`. Hierarchical dimensions (e.g. Wales → local authorities → wards) are flattened into this data frame: `parent` holds the reference of the parent value (`NA` at the top level) and `level` is the 1-based depth. Any value's `reference` can be used in a filter.

Returns `NULL` if the request fails.

Examples

```
## Not run:
datasets <- statswales_list_datasets()
filters <- statswales_get_filters(datasets$id[1])

# See available values for the first dimension
filters[[1]]$columnName
filters[[1]]$values

# Use a filter value in a data query
df <- statswales_get_dataset(
  datasets$id[1],
  filter = list(list(
    Year = filters[[1]]$values$reference[1:2]
  ))
)

## End(Not run)
```

`statswales_get_metadata` *Get metadata for a dataset*

Description

Returns full metadata for a published dataset including revision history, publication dates, and dimension information.

Usage

```
statswales_get_metadata(dataset_id, lang = "en-gb")
```

Arguments

`dataset_id`	A dataset UUID string. Use `statswales_list_datasets()` to find dataset IDs.
`lang`	Language for returned text. One of `"en-gb"` (default), `"en"`, `"cy-gb"`, or `"cy"`.

Value

A list containing dataset metadata. Returns `NULL` if the request fails or the dataset is not found.

Examples

```
## Not run:
datasets <- statswales_list_datasets()
meta <- statswales_get_metadata(datasets$id[1])

## End(Not run)
```

`statswales_get_pivot` *Retrieve a pivot table for a dataset*

Description

Returns a cross-tabulated view of dataset data. Choose one dimension for the columns (`x`) and one for the rows (`y`). The API stores the pivot configuration as a reusable query.

Usage

```
statswales_get_pivot(
  dataset_id,
  x,
  y,
  lang = "en-gb",
  page_number = 1,
  page_size = 100,
  filter = NULL,
  options = list(use_raw_column_names = FALSE, use_reference_values = FALSE,
    data_value_type = "formatted"),
  sort_by = NULL
)
```

Arguments

`dataset_id`	A dataset UUID string. Use `statswales_list_datasets()` to find dataset IDs.
`x`	Column name for the horizontal axis (column headers).
`y`	Column name for the vertical axis (row labels).
`lang`	Language for returned text. One of `"en-gb"` (default), `"en"`, `"cy-gb"`, or `"cy"`.
`page_number`	Page number to return. Default `1`.
`page_size`	Rows per page. Default `100`.
`filter`	Filter criteria in the same format as `statswales_get_dataset()`.
`options`	Display options in the same format as `statswales_get_dataset()`.
`sort_by`	Optional sort order, in the same format as `statswales_get_dataset()`.

Value

A list containing the paginated pivot view returned by the API, or `NULL` if the request fails.

Examples

```
## Not run:
datasets <- statswales_list_datasets()
id <- datasets$id[1]
filters <- statswales_get_filters(id)
```

```
pivot <- statswales_get_pivot(
  id,
  x = filters[[1]]$columnName,
  y = filters[[2]]$columnName
)

## End(Not run)
```

`statswales_get_query` *Inspect a stored query configuration*

Description

Returns the full configuration of a stored query, including the applied filters, total row count, column mappings, and the underlying SQL. The `filter_id` is returned by any call to `statswales_get_dataset()` (via the `POST /data` step) or `statswales_get_pivot()`.

Usage

```
statswales_get_query(dataset_id, filter_id)
```

Arguments

`dataset_id`	A dataset UUID string.
`filter_id`	A 12-character query identifier returned by the API.

Details

To retrieve the `filter_id` directly, call `statswales_create_query()`.

Value

A list containing the stored query details:

`id` The filter ID.

`totalLines` Total number of rows matching the query.

`requestObject` The filter and options that were submitted.

`columnMapping` Mapping between internal column names and display names, per language.

`query` A named list of SQL strings keyed by language code.

Returns `NULL` if the request fails.

Examples

```
## Not run:
datasets <- statswales_list_datasets()
id <- datasets$id[1]

fid <- statswales_create_query(id)
query_info <- statswales_get_query(id, fid)
query_info$totalLines

## End(Not run)
```

`statswales_get_topic` *Get the contents of a topic*

Description

Returns sub-topics or datasets tagged directly to a given topic. Use `statswales_list_topics()` to discover topic IDs.

Usage

```
statswales_get_topic(
  topic_id,
  lang = "en-gb",
  page_number = 1,
  page_size = 100,
  sort_by = NULL
)
```

Arguments

`topic_id`	A numeric topic ID. Use `statswales_list_topics()` to find IDs.
`lang`	Language for returned text. One of `"en-gb"` (default), `"en"`, `"cy-gb"`, or `"cy"`.
`page_number`	Page number to return. Default `1`.
`page_size`	Number of results per page. Default `100`.
`sort_by`	Optional sort order for datasets under a leaf topic. Either a string in `"column:direction"` format (e.g. `"title:asc"`) or a named character vector such as `c(title = "asc", last_updated_at = "desc")`.

Details

The returned list has four fields:

`selectedTopic` Details of the requested topic.

`children` Sub-topics under this topic. Empty for leaf topics.

`parents` Ancestor topics from root to the selected topic.

`datasets` A list with `data` (dataset list items) and `count`. Only populated for leaf topics (those with no `children`).

Value

A list with fields `selectedTopic`, `children`, `parents`, and `datasets`. Returns `NULL` if the request fails.

Examples

```
## Not run:
topics <- statswales_list_topics()
topic <- statswales_get_topic(topics$id[1])

# Sub-topics (non-leaf)
topic$children

# Datasets at a leaf topic
topic$datasets$data
topic$datasets$count

## End(Not run)
```

`statswales_list_datasets`

List all published datasets

Description

Returns a data frame of all published datasets available from the StatsWales public API. All pages are fetched automatically, so the full catalogue is returned.

Usage

```
statswales_list_datasets(lang = "en-gb")
```

Arguments

`lang`	Language for returned text. One of `"en-gb"` (default), `"en"`, `"cy-gb"`, or `"cy"`.

Value

A data frame with columns `id`, `title`, `first_published_at`, `last_updated_at`, and `archived_at`. Timestamp columns are `POSIXct` (UTC). Returns `NULL` if the request fails.

Examples

```
## Not run:
datasets <- statswales_list_datasets()
datasets_cy <- statswales_list_datasets(lang = "cy-gb")

## End(Not run)
```

`statswales_list_topics` *List top-level topics*

Description

Returns a data frame of all top-level topics that have at least one published dataset tagged against them.

Usage

```
statswales_list_topics(lang = "en-gb")
```

Arguments

`lang`	Language for returned text. One of `"en-gb"` (default), `"en"`, `"cy-gb"`, or `"cy"`.

Value

A data frame with columns `id`, `path`, `name`, `name_en`, and `name_cy`. Returns `NULL` if the request fails.

Examples

```
## Not run:
topics <- statswales_list_topics()

## End(Not run)
```

`statswales_search` *Search published datasets*

Description

Full-text search across dataset titles and summaries. Returns results ranked by relevance.

Usage

```
statswales_search(
  keywords,
  lang = "en-gb",
  page_number = 1,
  page_size = 100,
  mode = "basic"
)
```

Arguments

`keywords`	Search query string.
`lang`	Language for returned text. One of `"en-gb"` (default), `"en"`, `"cy-gb"`, or `"cy"`.
`page_number`	Page number to return. Default `1`.
`page_size`	Number of results per page. Default `100`.
`mode`	Search algorithm. One of `"basic"` (default), `"basic_split"`, `"fts"`, `"fts_simple"`, or `"fuzzy"`. The `fts` and `fts_simple` modes return highlighted matches in `match_title` and `match_summary` columns.

Value

A data frame with columns `id`, `title`, `summary`, `first_published_at`, `last_updated_at`, and `archived_at`. Timestamp columns are `POSIXct` (UTC). The `fts` and `fts_simple` modes also return `rank`, `match_title`, and `match_summary`. Returns `NULL` if the request fails.

Examples

```
## Not run:
results <- statswales_search("hospital")
results_cy <- statswales_search("ysbyty", lang = "cy-gb")
results_fuzzy <- statswales_search("transprt", mode = "fuzzy")

## End(Not run)
```

Index

statswales_create_query, 2
statswales_create_query(), *9*
statswales_download_dataset, 3
statswales_download_dataset(), *2*, *6*
statswales_get_dataset, 4
statswales_get_dataset(), *2, 3*, *6*, *8, 9*
statswales_get_filters, 6
statswales_get_filters(), *5*
statswales_get_metadata, 7
statswales_get_pivot, 8
statswales_get_pivot(), *6*, *9*
statswales_get_query, 9
statswales_get_query(), *2*
statswales_get_topic, 10
statswales_list_datasets, 11
statswales_list_datasets(), *2, 3*, *5–8*
statswales_list_topics, 12
statswales_list_topics(), *10*
statswales_search, 13