Package 'BFS'

October 20, 2026

Type Package

Title Get Data from the Swiss Federal Statistical Office

Version 0.8.0

Maintainer Felix Luginbuhl <felix.luginbuhl@protonmail.ch>

Description Search and download data from the Swiss Federal Statistical Office (BFS) APIs <https://www.bfs.admin.ch/>.

License GPL (>= 3)

Depends R (>= 4.0.0)

Imports pxweb, httr2 (>= 1.0.0), xml2, rvest, tibble, magrittr (>= 2.0.0), janitor, purrr (>= 1.0.0), dplyr, curl, zip, fs, tools, lifecycle, rlang (>= 1.0.0), tidyr

Suggests knitr, rmarkdown, testthat (>= 3.1.7), rstac, sf

Encoding UTF-8

RoxygenNote 7.3.3

URL https://felixluginbuhl.com/BFS/, https://github.com/lgnbhl/BFS/

BugReports https://github.com/lgnbhl/BFS/issues

Config/testthat/edition 3

LazyData true

VignetteBuilder knitr

NeedsCompilation no

Author Felix Luginbuhl [aut, cre, cph] (ORCID: <https://orcid.org/0009-0008-6625-2899>), Marius Hodel [ctb] (ORCID: <https://orcid.org/0000-0001-8789-506X>), Janosch Brenzel-Weiss [ctb], Joao Martins [ctb], Philipp Baumann [ctb] (ORCID: <https://orcid.org/0000-0002-3194-8975>)

Repository CRAN

Date/Publication 2026-10-20 17:30:02 UTC

Contents

bfs_clear_cache . . . 2
bfs_download_asset . . . 3
bfs_download_geodata . . . 4
bfs_get_asset_metadata . . . 5
bfs_get_base_maps . . . 6
bfs_get_catalog_data . . . 7
bfs_get_catalog_geodata . . . 9
bfs_get_catalog_tables . . . 10
bfs_get_data . . . 12
bfs_get_data_comments . . . 13
bfs_get_metadata . . . 14
bfs_get_sse_data . . . 15
bfs_get_sse_metadata . . . 17
bfs_sse_data . . . 18
bfs_sse_datasets . . . 20
bfs_sse_dimensions . . . 21
register_bzn . . . 23
register_dic . . . 23
register_gde . . . 24
register_gde_other . . . 25
register_kt . . . 25
register_kt_seeanteile . . . 26

Index **27**

`bfs_clear_cache` *Clear the BFS cache*

Description

Delete the base maps cached by `bfs_get_base_maps()` in `tools::R_user_dir(package = "BFS")` and the list of Swiss Stats Explorer datasets cached for the R session by `bfs_get_sse_data()` and `bfs_get_sse_metadata()`. The cached data is downloaded again when needed. In a long-running R session, e.g. a Shiny app, call it so that these two functions use the new version of a dataset. `bfs_sse_datasets()`, `bfs_sse_dimensions()` and `bfs_sse_data()` don't cache anything.

Usage

```
bfs_clear_cache()
```

Value

Invisibly, the path of the deleted cache folder.

Examples

```
bfs_clear_cache()
```

`bfs_download_asset` *Download a BFS file*

Description

Download any file of the BFS (Excel, CSV, PX, PDF, zip, ...) from the DAM API (https://dam-api.bfs.admin.ch/hub/swagger-ui/index.html) with its asset number or its BFS number. Use it for the tables of `bfs_get_catalog_tables()`, or for datasets too large for `bfs_get_data()`.

Usage

```
bfs_download_asset(
  number_asset = NULL,
  number_bfs = NULL,
  destfile,
  quiet = TRUE,
  mode = "wb",
  handle = curl::new_handle()
)
```

Arguments

`number_asset`	The asset number of the file, e.g. `"24025646"`. It is the `number_asset` column of `bfs_get_catalog_tables()` and `bfs_get_catalog_data()`, and the number at the end of the "assetdetail" URLs of the BFS website.
`number_bfs`	The BFS number of the file, e.g. `"px-x-1502040100_131"`. Use either `number_asset` or `number_bfs`.
`destfile`	Path of the downloaded file.
`quiet`	If `FALSE`, show the download progress.
`mode, handle`	Passed to `curl::curl_download()`.

Value

The path of the downloaded file. Returns `NULL` if there is no internet connection.

See Also

`bfs_get_asset_metadata()`, `bfs_get_catalog_tables()`

Examples

```
# a PX file, read it with pxR::read.px()
bfs_download_asset(
  number_bfs = "px-x-1502040100_131",
  destfile = file.path(tempdir(), "px-x-1502040100_131.px")
)
```

`bfs_download_geodata` *Download geodata from the Swiss geo-portal*

Description

Download the files of a geodata collection from the STAC API of the Swiss geo-portal (https://data.geo.admin.ch/api/stac/v0.9/). Requires the rstac package.

Usage

```
bfs_download_geodata(
  collection_id,
  output_dir = tempdir(),
  overwrite = FALSE,
  create_json = FALSE,
  bbox = NULL,
  asset_names = NULL,
  datetime = NULL,
  feature_id = NULL
)
```

Arguments

`collection_id`	The collection id, i.e. the `collection_id` column of `bfs_get_catalog_geodata()`, e.g. `"ch.bfs.generalisierte-grenzen_agglomerationen_g1"`.
`output_dir`	Folder of the downloaded files, a temporary folder by default.
`overwrite`	If `TRUE`, overwrite existing files.
`create_json`	If `TRUE`, also save the STAC items as a JSON file.
`bbox, datetime, feature_id`	Optional filters of the items: bounding box, date-time or interval, and item id. See `rstac::items()`.
`asset_names`	Optional names of the assets to download. See `rstac::assets_download()`.

Details

The use of the data is free but subject to the terms of use of the Swiss geo-portal (https://www.geo.admin.ch/terms-of-use).

Value

The paths of the downloaded files. Zip files are not extracted. Returns NULL if there is no internet connection.

See Also

bfs_get_catalog_geodata()

Examples

```
files <- bfs_download_geodata(
  collection_id = "ch.bfs.generalisierte-grenzen_agglomerationen_g1"
)
files
```

---

bfs_get_asset_metadata

Get the metadata of a BFS file

Description

Get the metadata of a BFS file from the DAM API (https://dam-api.bfs.admin.ch/hub/swagger-ui/index.html) with its asset number or its BFS number: title, description, publication date, contact, links, ...

Usage

```
bfs_get_asset_metadata(
  number_asset = NULL,
  number_bfs = NULL,
  language = c("de", "fr", "it", "en")
)
```

Arguments

number_asset	The asset number of the file, e.g. "24025646". It is the number_asset column of bfs_get_catalog_tables() and bfs_get_catalog_data(), and the number at the end of the "assetdetail" URLs of the BFS website.
number_bfs	The BFS number of the file, e.g. "px-x-1502040100_131". Use either number_asset or number_bfs.
language	Language of the metadata: "de" (default), "fr", "it" or "en".

Value

A nested list with the metadata. Returns NULL if there is no internet connection.

See Also

bfs_download_asset()

Examples

```
metadata <- bfs_get_asset_metadata(number_bfs = "px-x-1502040100_131", language = "en")
metadata$description$titles$main
```

---

bfs_get_base_maps *Get the Swiss base maps*

Description

Get the cartographic base maps of the BFS ThemaKart project: the borders of Switzerland, cantons, districts and communes, lakes, rivers, ... The names of the geometries and files are in the BFS documentation (https://www.bfs.admin.ch/bfs/en/home/statistics/regional-statistics/base-maps/cartographic-bases.html). Please read its terms of use and copyright notices.

Usage

```
bfs_get_base_maps(
  geom = NULL,
  category = "gf",
  type = "Poly",
  date = NULL,
  most_recent = TRUE,
  format = "shp",
  asset_number = "30566934",
  return_sf = TRUE
)
```

Arguments

geom	Geometry, e.g. "suis" (Switzerland), "kant" (cantons), "bezk" (districts), "polg" (communes), "seen" (lakes) or "flus" (rivers).
category	Category: "gf" or "total_area" (total area, default), "vf" or "vegetation_area" (vegetation area), or another part of the file name, e.g. "11" for the lakes.
type	"Poly" (polygons, default) or "Pnts" (points). Use "" for files without type, e.g. the rivers.
date	Optional date of validity, as "yyyymmdd", e.g. "20230101".
most_recent	If TRUE (default) and date is NULL, get the most recent file. If FALSE, get the first file found.
format	File format, "shp" (shapefile) by default.
asset_number	Asset number of the ThemaKart zip file. The default is the 2024 edition; e.g. "11927607" is the 2020 edition.

`return_sf` If `TRUE` (default), return an sf object (requires the sf package). If `FALSE`, return the path of the file. With `date = ""`, return the paths of all the matching files: use it, with `""` for the other arguments, to list the available files.

Details

The base maps are a zip file of several hundred files, downloaded once and cached in `tools::R_user_dir("BFS")`. Use `bfs_clear_cache()` to delete it. Each file name combines the arguments, e.g. `K4kant20220101gf_ch2007Poly.shp`: geometry `kant`, date `20220101`, category `gf`, type `Poly`, format `shp`.

Value

An sf object, or with `return_sf = FALSE` the file path(s). Returns `NULL` if there is no internet connection and the base maps are not cached yet.

See Also

`bfs_clear_cache()`

Examples

```
cantons <- bfs_get_base_maps(geom = "kant")
plot(sf::st_geometry(cantons))

lakes <- bfs_get_base_maps(geom = "seen", category = "11")
rivers <- bfs_get_base_maps(geom = "flus", category = "22", type = "")

# list all river files
bfs_get_base_maps(geom = "flus", category = "", type = "", date = "", return_sf = FALSE)
```

---

bfs_get_catalog_data *Search the BFS data catalog*

Description

Search the datasets of the BFS catalog (https://www.bfs.admin.ch/bfs/en/home/statistics/catalogue.html, article type "Data") that can be downloaded with `bfs_get_data()`.

Usage

```
bfs_get_catalog_data(
  language = "de",
  title = NULL,
  extended_search = NULL,
  spatial_division = NULL,
  prodima = NULL,
  inquiry = NULL,
  institution = NULL,
```

```
  publishing_year_start = NULL,
  publishing_year_end = NULL,
  order_nr = NULL,
  limit = 1000,
  article_model = 900033,
  article_model_group = 900029,
  return_raw = FALSE
)
```

Arguments

`language`	Language of the catalog: `"de"` (default), `"fr"`, `"it"` or `"en"`.
`title`	Text to search in the titles.
`extended_search`	Text to search in the titles, BFS number and summaries.
`spatial_division`	Spatial division of the datasets: `"Switzerland"`, `"Cantons"`, `"Districts"`, `"Communes"`, `"Other spatial divisions"` or `"International"`.
`prodima`	Theme number(s) of the BFS, e.g. `900212` for "Education and science". See the "Data catalog and PXWeb datasets" article for the list.
`inquiry`	Inquiry (survey) id.
`institution`	Institution id.
`publishing_year_start`, `publishing_year_end`	First and last publishing year.
`order_nr`	BFS number, e.g. `"px-x-1502040100_131"`.
`limit`	Maximum number of results (1000 by default). A message tells you when the catalog has as many results as `limit`.
`article_model`, `article_model_group`	Article type ids of the DAM API. The defaults select the "Data" (or "Tables") article type; you don't need to change them.
`return_raw`	If `TRUE`, return the catalog as returned by the API, with all its metadata in nested data frames.

Details

The English and Italian catalogs have fewer datasets than the German and French ones. Arguments accept several values, e.g. `prodima = c(900210, 900212)`.

Value

A tibble with the columns:

title Title of the dataset.

language Language of the catalog.

number_bfs BFS number, to use in `bfs_get_data()`.

number_asset Asset number, to use in `bfs_download_asset()`.

publication_date Publication date.

url_px URL of the dataset in the PXWeb API.

Returns NULL if there is no internet connection.

See Also

bfs_get_data(), bfs_get_catalog_tables()

Examples

```
bfs_get_catalog_data(language = "en", extended_search = "student")

# by theme ("Education and science")
bfs_get_catalog_data(language = "en", prodima = 900212)
```

---

bfs_get_catalog_geodata

Get the geodata catalog of the Swiss Confederation

Description

List the geodata collections of the STAC API of the Swiss geo-portal (https://data.geo.admin.ch/), from the BFS and other federal offices. Download a collection with bfs_download_geodata().

Usage

```
bfs_get_catalog_geodata(include_metadata = TRUE)
```

Arguments

include_metadata

If TRUE (default), add the metadata of each collection (title, description, dates, license, ...). This sends one request per collection (several hundred), so it takes a few minutes: save the result rather than calling it often. If FALSE, return only the collection ids and URLs.

Value

A tibble with one row per collection: collection_id, type and href, and with include_metadata = TRUE, title, description, created, updated, crs, license, provider_name, bbox and inverval. Returns NULL if there is no internet connection.

See Also

bfs_download_geodata()

Examples

```
catalog_geodata <- bfs_get_catalog_geodata()

# the collections of the BFS
catalog_geodata[grepl("^ch.bfs", catalog_geodata$collection_id), ]
```

`bfs_get_catalog_tables` *Search the BFS tables catalog*

Description

Search the tables of the BFS catalog (https://www.bfs.admin.ch/bfs/en/home/statistics/catalogue.html, article type "Tables"). Tables are mostly Excel or CSV files, not available in the PXWeb API: download them with `bfs_download_asset()`.

Usage

```
bfs_get_catalog_tables(
  language = "de",
  title = NULL,
  extended_search = NULL,
  spatial_division = NULL,
  prodima = NULL,
  inquiry = NULL,
  institution = NULL,
  publishing_year_start = NULL,
  publishing_year_end = NULL,
  order_nr = NULL,
  limit = 1000,
  article_model = 900030,
  article_model_group = 900029,
  return_raw = FALSE
)
```

Arguments

`language`	Language of the catalog: `"de"` (default), `"fr"`, `"it"` or `"en"`.
`title`	Text to search in the titles.
`extended_search`	Text to search in the titles, BFS number and summaries.
`spatial_division`	Spatial division of the datasets: `"Switzerland"`, `"Cantons"`, `"Districts"`, `"Communes"`, `"Other spatial divisions"` or `"International"`.
`prodima`	Theme number(s) of the BFS, e.g. `900212` for "Education and science". See the "Data catalog and PXWeb datasets" article for the list.

`inquiry`	Inquiry (survey) id.
`institution`	Institution id.
`publishing_year_start`, `publishing_year_end`	First and last publishing year.
`order_nr`	BFS number, e.g. `"px-x-1502040100_131"`.
`limit`	Maximum number of results (1000 by default). A message tells you when the catalog has as many results as `limit`.
`article_model`, `article_model_group`	Article type ids of the DAM API. The defaults select the "Data" (or "Tables") article type; you don't need to change them.
`return_raw`	If `TRUE`, return the catalog as returned by the API, with all its metadata in nested data frames.

Details

The English and Italian catalogs have fewer tables than the German and French ones. Arguments accept several values, e.g. `prodima = c(900210, 900212)`.

Value

A tibble with the columns:

title Title of the table.

language Language of the catalog.

number_asset Asset number, to use in `bfs_download_asset()`.

publication_date Publication date.

order_nr BFS number.

Returns `NULL` if there is no internet connection.

See Also

`bfs_download_asset()`, `bfs_get_catalog_data()`

Examples

```
tables <- bfs_get_catalog_tables(language = "en", extended_search = "students")
tables

# download the first table
bfs_download_asset(
  number_asset = tables$number_asset[1],
  destfile = file.path(tempdir(), "table.xlsx")
)
```

bfs_get_data *Get a BFS dataset*

Description

Download a dataset from the PXWeb API of the BFS with its BFS number. Find the BFS number with `bfs_get_catalog_data()` or on the BFS website.

Usage

```
bfs_get_data(
  number_bfs,
  language = "de",
  query = NULL,
  column_name_type = "text",
  variable_value_type = "text",
  clean_names = FALSE,
  delay = NULL
)
```

Arguments

`number_bfs`	The BFS number of a dataset, e.g. `"px-x-1502040100_131"`.
`language`	Language of the dataset: `"de"` (default), `"fr"`, `"it"` or `"en"`.
`query`	Optional selection of the data: a named list with the variable codes as names and the value codes as values (columns `code` and `values` of `bfs_get_metadata()`), e.g. `list(Jahr = c("40", "41"), Geschlecht = "*")`. `"*"` selects all values. A variable not in `query` is left out of the result (totals over it), if the dataset allows it. If `NULL` (default), the whole dataset is downloaded.
`column_name_type`	Column names as `"text"` (label, default) or `"code"`.
`variable_value_type`	Values as `"text"` (label, default) or `"code"`.
`clean_names`	If `TRUE`, clean the column names with `janitor::clean_names()`.
`delay`	Optional number of seconds to wait before downloading the data, to avoid "Too Many Requests" errors when calling the API in a loop.

Details

Large datasets can exceed the limits of the API: select only the data you need with `query`, or download the whole file with `bfs_download_asset()`.

The PXWeb API is the API of STAT-TAB, which the BFS is replacing with the Swiss Stats Explorer: check first if the statistic is on `https://stats.swiss/` and use `bfs_sse_data()`.

Value

A tibble. Returns NULL if there is no internet connection.

See Also

bfs_get_catalog_data(), bfs_get_metadata(), bfs_get_data_comments()

Examples

```
# whole dataset
bfs_get_data(number_bfs = "px-x-1502040100_131", language = "en")

# only some years and levels of study
bfs_get_data(
  number_bfs = "px-x-1502040100_131",
  language = "en",
  query = list(Jahr = c("40", "41"), Studienstufe = c("2", "3"))
)
```

---

bfs_get_data_comments *Get the comments of a BFS dataset*

Description

Get the comments (footnotes) of a dataset of the PXWeb API, e.g. on provisional values or changes of definition.

Usage

```
bfs_get_data_comments(
  number_bfs,
  language = "de",
  query = NULL,
  clean_names = FALSE,
  delay = NULL
)
```

Arguments

number_bfs	The BFS number of a dataset, e.g. "px-x-1502040100_131".
language	Language of the dataset: "de" (default), "fr", "it" or "en".
query	Optional selection of the data: a named list with the variable codes as names and the value codes as values (columns code and values of bfs_get_metadata()), e.g. list(Jahr = c("40", "41"), Geschlecht = "*"). "*" selects all values. A variable not in query is left out of the result (totals over it), if the dataset allows it. If NULL (default), the whole dataset is downloaded.

clean_names	If TRUE, clean the column names with janitor::clean_names().
delay	Optional number of seconds to wait before downloading the data, to avoid "Too Many Requests" errors when calling the API in a loop.

Details

The PXWeb API is the API of STAT-TAB, which the BFS is replacing with the Swiss Stats Explorer.

Value

A tibble with one row per comment. Returns NULL if there is no internet connection.

See Also

bfs_get_data()

Examples

```
bfs_get_data_comments(number_bfs = "px-x-1502040100_131", language = "en")
```

---

bfs_get_metadata *Get the variables of a BFS dataset*

Description

Get the variables of a dataset of the PXWeb API and the values each variable can take. Use them to build the query of bfs_get_data().

Usage

```
bfs_get_metadata(number_bfs, language = "de")
```

Arguments

number_bfs	The BFS number of a dataset, e.g. "px-x-1502040100_131".
language	Language of the dataset: "de" (default), "fr", "it" or "en".

Details

The PXWeb API is the API of STAT-TAB, which the BFS is replacing with the Swiss Stats Explorer (see bfs_sse_dimensions()).

Value

A tibble with one row per variable and the columns code and text (variable code and label), values and valueTexts (list columns with the value codes and labels) and title (title of the dataset). Use tidyr::unnest_longer(metadata, c(values, valueTexts)) to get one row per value. Returns NULL if there is no internet connection.

See Also

bfs_get_data()

Examples

```
metadata <- bfs_get_metadata(number_bfs = "px-x-1502040100_131", language = "en")
metadata

# one row per value
tidyr::unnest_longer(metadata, c(values, valueTexts))
```

---

bfs_get_sse_data *Get data from the Swiss Stats Explorer (deprecated)*

Description

[Deprecated]

Deprecated since BFS 0.8.0: use bfs_sse_data() instead. It downloads the data faster and stops on a query that wouldn't filter the data. This function still works and gets only bug fixes, e.g. the labels of codes with no name in language (see NEWS). To move your code, see the article Moving to the new Swiss Stats Explorer functions.

Download a dataset of the Swiss Stats Explorer (SSE) API (https://stats.swiss/). Select the data you need with query, built from the codelists of bfs_get_sse_metadata().

Usage

```
bfs_get_sse_data(
  number_bfs,
  language = "de",
  start_period = NULL,
  end_period = NULL,
  query = NULL,
  column_name_type = "code",
  variable_value_type = "text",
  clean_names = FALSE
)
```

Arguments

number_bfs	The id of an SSE dataset (dataflow), e.g. "DF_LWZ_1". It is shown in the URL of the dataset on https://stats.swiss/.
language	Language of the labels: "de" (default), "fr", "it" or "en".
start_period, end_period	Optional first and last period of the data, e.g. "2020".

query
: A named list: the names are dimension codes (column code of `bfs_get_sse_metadata()`) and the values are value codes (column `value`), e.g. `list(FREQ = "M", COUNTRY_ORIGIN = c("CH", "AUSL"))`. A dimension not in `query` keeps all its values. If `NULL` (default), the whole dataset is downloaded, which can be large and slow.

column_name_type
: Column names as dimension `"code"` (default) or `"text"` (label).

variable_value_type
: Values as `"text"` (label, default) or `"code"`.

clean_names
: If `TRUE`, clean the column names with `janitor::clean_names()`.

Details

The list of the SSE datasets is cached for the R session. A long-running R session, e.g. a Shiny app, keeps using the version of a dataset it saw first: call `bfs_clear_cache()` to use a new version, or use `bfs_sse_data()`, which caches nothing.

Value

A tibble with one column per dimension and the observations in the `value` column. Returns `NULL` if there is no internet connection.

Author(s)

Marius Hodel

See Also

`bfs_sse_data()`, `bfs_get_sse_metadata()`

Examples

```
bfs_get_sse_data(
  number_bfs = "DF_PASTA_552_MONTHLY",
  language = "en",
  query = list(
    FREQ = "M",
    ACCOMMODATION_TYPE = "552001",
    COUNTRY_ORIGIN = c("CH", "AUSL")
  ),
  start_period = "2020",
  end_period = "2023"
)
```

`bfs_get_sse_metadata` *Get the codelists of a Swiss Stats Explorer dataset (deprecated)*

Description

[Deprecated]

Deprecated since BFS 0.8.0: use `bfs_sse_dimensions()` instead. This function still works and gets only bug fixes, e.g. the labels of codes with no name in `language` (see NEWS). To move your code, see the article Moving to the new Swiss Stats Explorer functions.

Get the dimensions of a dataset of the Swiss Stats Explorer (SSE) API (`https://stats.swiss/`) and the values each dimension can take. Use them to build the `query` of `bfs_get_sse_data()`.

Usage

```
bfs_get_sse_metadata(number_bfs, language = "de")
```

Arguments

`number_bfs`	The id of an SSE dataset (dataflow), e.g. `"DF_LWZ_1"`. It is shown in the URL of the dataset on `https://stats.swiss/`.
`language`	Language of the labels: `"de"` (default), `"fr"`, `"it"` or `"en"`. A label missing in this language falls back to another language.

Details

The list of the SSE datasets is cached for the R session. A long-running R session, e.g. a Shiny app, keeps using the version of a dataset it saw first: call `bfs_clear_cache()` to use a new version, or use `bfs_sse_dimensions()`, which caches nothing.

Value

A tibble with one row per value of each dimension:

code The dimension code, to use as a name in `query`.

text The dimension label.

value The value code, to use as a value in `query`.

valueText The value label.

position_dimension The position of the dimension in the dataset.

Returns `NULL` if there is no internet connection.

Author(s)

Marius Hodel

See Also

bfs_sse_dimensions(), bfs_get_sse_data()

Examples

```
codelist <- bfs_get_sse_metadata("DF_LWZ_1", language = "en")
codelist

# the dimensions and their codes
unique(codelist[c("code", "text")])
```

---

bfs_sse_data *Get data from the Swiss Stats Explorer*

Description

Download the data of a dataset of the Swiss Stats Explorer (SSE) API (https://stats.swiss/). Select the data you need with filters, built from bfs_sse_dimensions().

Usage

```
bfs_sse_data(
  dataset,
  ...,
  filters = NULL,
  start_period = NULL,
  end_period = NULL,
  first_n = NULL,
  last_n = NULL,
  language = "de",
  labels = c("both", "code", "label"),
  attributes = FALSE,
  version = NULL
)
```

Arguments

dataset	The id of a dataset, e.g. "DF_LWZ_1": the column dataset of bfs_sse_datasets(), also shown in the URL of the dataset on https://stats.swiss/.
...	These dots are for future extensions and must be empty: the arguments after dataset must be named. The arguments of bfs_get_sse_data(), such as number_bfs or query, give an error that names their new name.
filters	A named list: the names are dimensions (column dimension of bfs_sse_dimensions()) and the values are value codes (column value), e.g. list(FREQ = "M", COUNTRY_ORIGIN = c("CH", "AUSL")). A dimension not in filters keeps all its values. If NULL (default), the whole dataset is downloaded, which can be large and slow. An unknown dimension is an error, as the data wouldn't be filtered on it.

`start_period`, `end_period`
: Optional first and last period of the data, e.g. `"2020"` or `"2020-01"`.

`first_n`, `last_n`
: Optional number of first or last periods of each series (each combination of the dimensions), e.g. `last_n = 1` for the latest value. Sent to the API as the SDMX parameters `firstNObservations` and `lastNObservations`. They can be combined with `start_period` and `end_period`.

`language`
: Language of the labels: `"de"` (default), `"fr"`, `"it"` or `"en"`. A label missing in this language falls back to another language.

`labels`
: The values of the dimensions and attributes: `"both"` (default) to keep the codes and add a column `<dimension>_label` after each dimension, `"code"` for the codes only, or `"label"` for the labels only. Different codes can have the same label (e.g. the canton and the commune of Bern), so use `"label"` only to display the data.

`attributes`
: If `TRUE`, add the attributes of the observations after `value`, e.g. `OBS_STATUS` (status of the value, e.g. provisional). They can change from one version of a dataset to another.

`version`
: Version of the dataset, e.g. `"1.1.0"`: the column `version` of `bfs_sse_datasets()`. If `NULL` (default), the API uses the latest version. Set it in a data pipeline to keep the same structure when a new version is published.

Details

The data is downloaded in the SDMX-CSV format, with the labels in `language` given by the API. It is smaller and faster to read than the SDMX-ML format used by `bfs_get_sse_data()`.

The function sends two requests: one for the dimensions of the dataset, to build the query from `filters`, and one for the data. The values in `filters` are checked by the API: a value not in the dataset gives no data.

Value

A tibble with one column per dimension, named with the dimension id whatever the `language`, then `TIME_PERIOD` as text (e.g. `"2024"`, `"2024-Q1"`, `"2024-01"`), as given by the API, and the observations in the numeric `value` column. Returns `NULL` if there is no internet connection. The errors have a class, to handle them with `tryCatch()`:

`bfs_no_records` No data matches the query.

`bfs_sse_filter_error` `filters` wouldn't filter the data, e.g. an unknown dimension.

`bfs_sse_argument_error` Another argument is not valid.

`bfs_sse_dataset_error` The dataset or the version doesn't exist.

`bfs_sse_query_error` The API rejects the query, e.g. a period in a wrong format or a URL too long. The error gives the message of the API.

`bfs_http_error` The API is not available.

Author(s)

Felix Luginbuhl

See Also

bfs_sse_datasets(), bfs_sse_dimensions()

Examples

```
bfs_sse_data(
  "DF_PASTA_552_MONTHLY",
  filters = list(
    FREQ = "M",
    ACCOMMODATION_TYPE = "552001",
    COUNTRY_ORIGIN = c("CH", "AUSL")
  ),
  start_period = "2020",
  end_period = "2023",
  language = "en"
)

# the latest vacancy rate of each canton
bfs_sse_data(
  "DF_LWZ_1",
  filters = list(
    GR_KT_GDE = c("GE", "ZH", "BE"),
    WOHN_ANZAHL = "_T",
    LEERWOHN_TYP = "_T",
    MEASURE_DIMENSION = "RATE"
  ),
  last_n = 1,
  language = "en"
)
```

---

bfs_sse_datasets *List the datasets of the Swiss Stats Explorer*

Description

List the datasets (dataflows) of the Swiss Stats Explorer (SSE) API (https://stats.swiss/), to find the id of a dataset for bfs_sse_dimensions() and bfs_sse_data().

Usage

```
bfs_sse_datasets(language = "de")
```

Arguments

language	Language of the names: "de" (default), "fr", "it" or "en".

Value

A tibble with one row per version of each dataset, the latest version first:

dataset The id of the dataset, e.g. `"DF_LWZ_1"`.

name The name of the dataset.

version The version of the dataset.

agency The agency publishing the dataset.

description The description of the dataset version, or `NA`, as given by the API: it can contain HTML tags such as `<br>`.

Returns `NULL` if there is no internet connection.

Author(s)

Felix Luginbuhl

See Also

`bfs_sse_dimensions()`, `bfs_sse_data()`

Examples

```
datasets <- bfs_sse_datasets(language = "en")
datasets

# datasets about tourism, in their name or description
datasets[grepl("touris", paste(datasets$name, datasets$description), ignore.case = TRUE), ]
```

---

bfs_sse_dimensions *Get the dimensions of a Swiss Stats Explorer dataset*

Description

Get the dimensions of a dataset of the Swiss Stats Explorer (SSE) API (https://stats.swiss/) and the values allowed in each dimension. Use them to build the `filters` of `bfs_sse_data()`.

Usage

```
bfs_sse_dimensions(dataset, ..., language = "de", version = NULL)
```

Arguments

dataset	The id of a dataset, e.g. "DF_LWZ_1": the column dataset of bfs_sse_datasets(), also shown in the URL of the dataset on https://stats.swiss/.
...	These dots are for future extensions and must be empty: the arguments after dataset must be named. The arguments of bfs_get_sse_metadata(), such as number_bfs, give an error that names their new name.
language	Language of the labels: "de" (default), "fr", "it" or "en". A label missing in this language falls back to another language.
version	Version of the dataset, e.g. "1.1.0": the column version of bfs_sse_datasets(). If NULL (default), the API uses the latest version. Set it in a data pipeline to keep the same structure when a new version is published.

Value

A tibble with one row per value allowed in each dimension (some values may have no data):

dimension The dimension id, to use as a name in filters.

dimension_label The dimension label.

value The value code, to use as a value in filters.

value_label The value label.

position The position of the dimension in the dataset.

parent The code of the parent value in the codelist, e.g. the district of a commune and the canton of a district, or NA. Use it to select all the values under a value.

Returns NULL if there is no internet connection.

Author(s)

Felix Luginbuhl

See Also

bfs_sse_datasets(), bfs_sse_data()

Examples

```
dimensions <- bfs_sse_dimensions("DF_PASTA_552_MONTHLY", language = "en")
dimensions

# the dimensions only
unique(dimensions[c("dimension", "dimension_label")])

# the communes of the canton of Geneva: the values whose parent is a
# district of Geneva
regions <- bfs_sse_dimensions("DF_LWZ_1", language = "en")
districts <- regions$value[regions$parent %in% "GE"]
regions[regions$parent %in% districts, ]
```

register_bzn *Swiss official commune register: districts*

Description

The districts of Switzerland (or comparable administrative entities) on 1 January 2023.

Usage

```
register_bzn
```

Format

A data frame with 143 rows and 3 columns:

GDEKT Canton abbreviation, e.g. "ZH".
GDEBZNR District number.
GDEBZNA District name.

Source

https://www.bfs.admin.ch/bfs/en/home/basics/swiss-official-commune-register.html

See Also

Other commune register: register_dic, register_gde, register_gde_other, register_kt, register_kt_seeanteile

register_dic *Swiss official commune register: dictionary*

Description

The variables of the commune register datasets, in German, French and Italian.

Usage

```
register_dic
```

Format

A data frame with 27 rows and 5 columns:

language Language: "de", "fr" or "it".
abbreviation Variable name, e.g. "GDENR".
title Variable label.
spec Specifications.
notes General notes.

Source

https://www.bfs.admin.ch/bfs/en/home/basics/swiss-official-commune-register.html

See Also

Other commune register: register_bzn, register_gde, register_gde_other, register_kt, register_kt_seeanteile

---

register_gde *Swiss official commune register: communes*

Description

The communes of Switzerland on 1 January 2023, with their district and canton. The BFS assigns a number to each commune and publishes the Swiss official commune register, used by the federal, cantonal and communal administrations to identify communes. See register_dic for the variables in German, French and Italian.

Usage

```
register_gde
```

Format

A data frame with 2,136 rows and 8 columns:

GDEKT Canton abbreviation, e.g. "ZH".

GDEBZNR District number.

GDENR BFS commune number.

GDENAME Official commune name.

GDENAMK Short commune name.

GDEBZNA District name.

GDEKTNA Canton name.

GDEMUTDAT Date of the last change, as "dd.mm.yyyy".

Source

https://www.bfs.admin.ch/bfs/en/home/basics/swiss-official-commune-register.html

See Also

Other commune register: register_bzn, register_dic, register_gde_other, register_kt, register_kt_seeanteile

register_gde_other *Swiss official commune register: areas without a commune*

Description

The areas of Switzerland that belong to no commune (e.g. the Staatswald Galm), on 1 January 2023.

Usage

```
register_gde_other
```

Format

A data frame with 3 rows and 4 columns:

GDENR BFS number of the area.

GDENAME Name of the area.

KTNR Canton number.

GDEBZNR District number.

Source

https://www.bfs.admin.ch/bfs/en/home/basics/swiss-official-commune-register.html

See Also

Other commune register: register_bzn, register_dic, register_gde, register_kt, register_kt_seeanteile

register_kt *Swiss official commune register: cantons*

Description

The 26 cantons of Switzerland.

Usage

```
register_kt
```

Format

A data frame with 26 rows and 3 columns:

KTNR Canton number.

GDEKT Canton abbreviation, e.g. "ZH".

GDEKTNA Canton name.

Source

https://www.bfs.admin.ch/bfs/en/home/basics/swiss-official-commune-register.html

See Also

Other commune register: register_bzn, register_dic, register_gde, register_gde_other, register_kt_seeanteile

---

`register_kt_seeanteile`

Swiss official commune register: cantonal shares of lakes

Description

The lakes of Switzerland, split by canton (e.g. "Zürichsee (ZH)" and "Zürichsee (SZ)"), on 1 January 2023.

Usage

```
register_kt_seeanteile
```

Format

A data frame with 41 rows and 3 columns:

GDENR BFS number of the lake share.

GDENAME Name of the lake share.

KTNR Canton number.

Source

https://www.bfs.admin.ch/bfs/en/home/basics/swiss-official-commune-register.html

See Also

Other commune register: register_bzn, register_dic, register_gde, register_gde_other, register_kt

Index

∗ **commune register**
    register_bzn, 23
    register_dic, 23
    register_gde, 24
    register_gde_other, 25
    register_kt, 25
    register_kt_seeanteile, 26
∗ **datasets**
    register_bzn, 23
    register_dic, 23
    register_gde, 24
    register_gde_other, 25
    register_kt, 25
    register_kt_seeanteile, 26

bfs_clear_cache, 2
bfs_clear_cache(), *7*, *16, 17*
bfs_download_asset, 3
bfs_download_asset(), *6*, *8*, *10–12*
bfs_download_geodata, 4
bfs_download_geodata(), *9*
bfs_get_asset_metadata, 5
bfs_get_asset_metadata(), *3*
bfs_get_base_maps, 6
bfs_get_base_maps(), *2*
bfs_get_catalog_data, 7
bfs_get_catalog_data(), *3*, *5*, *11–13*
bfs_get_catalog_geodata, 9
bfs_get_catalog_geodata(), *4, 5*
bfs_get_catalog_tables, 10
bfs_get_catalog_tables(), *3*, *5*, *9*
bfs_get_data, 12
bfs_get_data(), *3*, *7–9*, *14, 15*
bfs_get_data_comments, 13
bfs_get_data_comments(), *13*
bfs_get_metadata, 14
bfs_get_metadata(), *12, 13*
bfs_get_sse_data, 15
bfs_get_sse_data(), *2*, *17–19*
bfs_get_sse_metadata, 17
bfs_get_sse_metadata(), *2*, *15, 16*, *22*
bfs_sse_data, 18
bfs_sse_data(), *2*, *12*, *15, 16*, *20–22*
bfs_sse_datasets, 20
bfs_sse_datasets(), *2*, *18–20*, *22*
bfs_sse_dimensions, 21
bfs_sse_dimensions(), *2*, *14*, *17, 18*, *20, 21*

curl::curl_download(), *3*

janitor::clean_names(), *12*, *14*, *16*

register_bzn, 23, *24–26*
register_dic, *23*, 23, *24–26*
register_gde, *23, 24*, 24, *25, 26*
register_gde_other, *23, 24*, 25, *26*
register_kt, *23–25*, 25, *26*
register_kt_seeanteile, *23–26*, 26
rstac::assets_download(), *4*
rstac::items(), *4*

tryCatch(), *19*