

Package: BFS (via r-universe)

October 17, 2026

Type Package

Title Get Data from the Swiss Federal Statistical Office

Version 0.7.3

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, janitor, purrr (>= 1.0.0), dplyr, curl, zip, fs, tools, lifecycle, rlang, tidyr

Suggests knitr, rmarkdown, testthat (>= 3.0.0), 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

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libxml2-dev libssl-dev

Repository <https://swissstatsr.r-universe.dev>

Date/Publication 2026-10-17 19:49:25 UTC

RemoteUrl <https://github.com/lgnbhl/BFS>

RemoteRef HEAD

RemoteSha 1a13a524ee69b3cc3cb462c5b2e7e48e72d92769

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	16
register_bzn	17
register_dic	18
register_gde	19
register_gde_other	20
register_kt	20
register_kt_seeanteile	21
Index	22

bfs_clear_cache	<i>Clear the BFS cache</i>
-----------------	----------------------------

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 dataflows cached for the R session by [bfs_get_sse_data()] and [bfs_get_sse_metadata()]. The cached data is downloaded again when needed.

Usage

```
bfs_clear_cache()
```

Value

Invisibly, the path of the deleted cache folder.

Examples

```
bfs_clear_cache()
```

bfs_download_asset	<i>Download a BFS file</i>
--------------------	----------------------------

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>>).
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), "bez" (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

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

<code>limit</code>	Maximum number of results (1000 by default). A message tells you when the catalog has as many results as 'limit'.
<code>article_model, article_model_group</code>	Article type ids of the DAM API. The defaults select the "Data" (or "Tables") article type; you don't need to change them.
<code>return_raw</code>	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_get_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")
```

<code>bfs_get_metadata</code>	<i>Get the variables of a BFS dataset</i>
-------------------------------	---

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_get_sse_metadata()]`).

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

Description

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()].

Value

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

See Also

[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*

Description

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.

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.

See Also

[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")])
```

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	<i>Swiss official commune register: communes</i>
--------------	--

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	<i>Swiss official commune register: areas without a commune</i>
--------------------	---

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	<i>Swiss official commune register: cantons</i>
-------------	---

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, [17](#)
- register_dic, [18](#)
- register_gde, [19](#)
- register_gde_other, [20](#)
- register_kt, [20](#)
- register_kt_seeanteile, [21](#)

* datasets

- register_bzn, [17](#)
- register_dic, [18](#)
- register_gde, [19](#)
- register_gde_other, [20](#)
- register_kt, [20](#)
- register_kt_seeanteile, [21](#)

- 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, [16](#)

- register_bzn, [17](#), [18–21](#)
- register_dic, [18](#), [18–21](#)
- register_gde, [18](#), [19](#), [20](#), [21](#)
- register_gde_other, [18](#), [19](#), [20](#), [21](#)
- register_kt, [18](#), [19](#), [20](#), [20](#), [21](#)
- register_kt_seeanteile, [18–20](#), [21](#), [21](#)