Package: swissgeo.core (via r-universe)

July 1, 2026

Type Package

Title Core Package to Access Data via the Geoportal des Bundes (Swiss Federal Government Open Data Portal)

Version 0.0-9001

Date 2026-06-29

Maintainer Reto Stauffer <reto.stauffer@uibk.ac.at>

Depends httr2, sf, parsedate, tidyr, dplyr, utils

Suggests tinytest, knitr

Description Package to download data from data.geo.admin.ch.

URL https://retostauffer.github.io/swissgeo.core

License GPL-2

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://retostauffer.r-universe.dev

Date/Publication 2026-07-01 18:30:43 UTC

RemoteUrl http://codeberg.org/retostauffer/swissgeo.core

RemoteRef HEAD

RemoteSha 8287a3c76782a9f51b0953fae6f0677895cb1467

Contents

`assets` *Assets*

Description

Each item can provide multiple assets. An asset specifies the type of data (e.g., historical, recent) as well as the temporal resolution of the data and the data period (if suitable). The `sg_assets` class is used to handle these assets within the package.

Usage

```
assets(x)

## S3 method for class 'assets'
format(x, ...)

## S3 method for class 'assets'
print(x, ...)

## S3 method for class 'assets'
x[i, ...]

## S3 method for class 'assets'
as.double(x, ...)

## S3 method for class 'assets'
names(x)

## S3 method for class 'assets'
as.data.frame(x, row.names = NULL, optional = NA, ...)
```

Arguments

`x`	object of class 'assets'.
`...`	currently ignored.
`i`	elements to extract or replace. Numeric values coerced to integer identifying the `i`th element(s) for subsetting.

row.names	NULL (default) or a character vector giving the row names for the data frame. Only used if x is of length 1L.
optional	currently ignored.

Value

An asset data frame with the name of the asset and all provided additional information.

Author(s)

Reto

---

autoconvert_datetime *Converting Datetime Columns*

Description

Takes a data frame as input and tries to identify columns containing datetime information.

Usage

```
autoconvert_datetime(x)
```

Arguments

x	data frame.

Details

Data received by the API or read via the files provided regularey contain date or datetime information as character strings.

This function takes a data frame and tries to identify variables/columns containing date and time information to coerce the information into Date or POSIXct objects. Currently, the following formats are checked currently:

- `2026-01-16T17:34:34(...)Z`: converted to POSIXct
- `16.01.2026`: converted to Date
- `2026-01-16`: converted to Date
- `16.01.2026 12:03`: converted to POSIXct w/ time zone Europe/Zuerich
- `16.01.2026 00:00`: converted to Date

Value

Data frame with POSIXct variables if any variable containing datetime information was detected.

Author(s)

Reto

Examples

```
## Not run:
autoconvert_datetime(data.frame(a = 1, b = "2026-01-16T12:34:56.23435Z"))
autoconvert_datetime(data.frame(a = 1, b = "2026-01-16T12:34:56Z"))
autoconvert_datetime(data.frame(a = 1, b = "16.01.2026 12:34"))
autoconvert_datetime(data.frame(a = 1, b = "16.01.2026 00:00"))
autoconvert_datetime(data.frame(a = 1, b = "16.01.2026"))
autoconvert_datetime(data.frame(a = 1, b = "2026-01-16"))

## End(Not run)
```

---

dataframe_to_sf *Convert Dataframe to Simple Feature Dataframe*

Description

Many data sets retrieved via the Geoportal are geolocated, providing coordinates of e.g., a station location. This function tries to auto-detect locations, converting a dataframe into a simple feature dataframe.

Usage

```
dataframe_to_sf(x, use_crs = c("lv95", "wgs84"), verbose = FALSE)
```

Arguments

x	a data frame which (potentially) contains coordinates.
use_crs	character, order of coordinate reference to use (if found).
verbose	logical, defaults to FALSE. If set TRUE the some messages are printed.

Details

Some data sets like station locations come their geolocation (coordinates), sometimes given in two different reference systems. This function takes a data frame and checks if it can find coordinates. If found, the object is converted into a simple feature data frame dropping all coordinates from the data part.

Value

If no matching coordinates are found, x is returned as is. Else x is converted into a simple feature data frame, removing all coordinate related columns.

remove_language_cols *Removing Non-required Language-specific Variables*

Description

Removing Non-required Language-specific Variables

Usage

```
remove_language_cols(x, lang)
```

Arguments

x	data frame to be checked and potentially modified.
lang	character, languages to keep. If "all" no modifications are done. Else trying to remove all columns not matching the requested lang.

Value

A data frame, potentially modified by removing all columns not matching the requested language.

Author(s)

Reto

sg_api_url *Generate Geoportal API URLs*

Description

Generate Geoportal API URLs

Usage

```
sg_api_url(req, ...)
```

Arguments

...	additional arguments to extend the base API URL. All arguments (after coercion to character) must represent valid non-empty characters.

Details

On loading the package sets the option "swissgeo.apiurl" with the main entry point to the API. This is used inside this function (i.e., by the entire package) to talk to the API.

This would allow users to change the end-point (or version) without updating the package, e.g., for testing.

Value

Character of length one with the URL of the API end point.

Author(s)

Reto

Examples

```
## Base URL
req <- getOption("swissgeo.req")
sg_api_url(req)$url

## Extending the base URL to point to a specific API endpoint
req |> sg_api_url("ch.meteoschweiz.ogd-smn")$url
req |> sg_api_url("ch.meteoschweiz.ogd-smn", "items")$url

## Changing default API URL. Possible, but not a standard use-case.
hold_apiurl <- getOption("swissgeo.apiurl") # kept for resetting
options(swissgeo.apiurl = "https://some.new.domain/api/stac/v5")
getOption("swissgeo.req") |> sg_api_url()$url

## Setting back to defaults
options(swissgeo.apiurl = hold_apiurl)
getOption("swissgeo.req") |> sg_api_url()$url
```

---

sg_assets *Collection Assets*

Description

Retrieving all assets of a specific collection.

Usage

```
sg_assets(id, verbose = FALSE, raw = FALSE)
```

Arguments

id	character, ID of the collection for which to retrieve the items (see sg_collections()).
verbose	logical, defaults to FALSE. If set TRUE the some messages are printed.
raw	logical, defaults to FALSE (see Return).

Value

A tbl data frame containing the asset details is returned.

If raw = TRUE a list of lists is returned with the raw (json decoded) result from the API calls. Each element in the list corresponds to one API call (paging).

Author(s)

Reto

`sg_collections` *Collections*

Description

Requesting a list of all available collections provided via the API.

Usage

```
sg_collections(pattern = NULL, verbose = FALSE, raw = FALSE)
```

Arguments

`pattern`	`NULL` or a pattern used return a subset of all available collections, applied to the collection ID.
`verbose`	logical, defaults to `FALSE`. If set `TRUE` the some messages are printed.
`raw`	logical, defaults to `FALSE` (see Return).

Value

Returns a tibble data frame with the available collections. If `pattern = NULL` all collections found are returned. If a `pattern` is specified, all collections where the collection ID (`id`) matches the pattern are returned. If no IDs match the pattern, a warning is thrown and `NULL` is returned.

If `raw = TRUE` is specified, `pattern` is ignored and the raw data as retrieved by the API are returned as a list of lists. Each element of the list corresponds to one API call containing a set of lists corresponding to the different collections returned.

Author(s)

Reto

Examples

```
## Not run:
## Fetch all available collections
collections <- sg_collections()

## Extract collections by MeteoSchweiz
subset(res, grepl("meteoschweiz", id))

## End(Not run)
```

`sg_download_asset` *Download (Cache) and Import Assets/Data Sets*

Description

Download (Cache) and Import Assets/Data Sets

Usage

```
sg_download_asset(
  x,
  dir = NULL,
  language = c("all", "en", "de", "fr", "it"),
  use_crs = c("lv95", "wgs84"),
  verbose = FALSE
)
```

Arguments

`x`	data frame (with one row) or list containing at least the following variables: id, type, href, file_checksum.
`dir`	character, name/path to a directory to cache the data sets (see 'Details' for more information).
`language`	character, one of `"all"` (default), `"en"`, `"de"`, `"fr"`, or `"it"`. If set different to `"all"` columns ending in _<lang> not matching the requested language will be removed.
`use_crs`	character vector used to auto-detect coordinates. Can be set to `NULL`/`FALSE` to not do the conversion.
`verbose`	logical, defaults to `FALSE`. If set `TRUE` the some messages are printed.

Details

Assets (data sets) provided via geo.admin.ch come with a file checksum which can be used to check if a file changed. We use this to cache files.

If a directory is specified (`dir`) it will be checked if that file has already been downloaded and stored in the directory. In this situation, the local file is used rather than downloading the data. If not, the asset will be downloaded and stored in the directory specified so that it can be potentially used again next time.

Please note that the the size of the directory can grow fast, especially if this mechanic is used when downloading/accessing data sets that rapidly change (i.e., most recent data) as a new file will be created every time the checksum changes.

Author(s)

Reto

sg_items *Collection Items*

Description

Retrieving all items of a specific collection.

Usage

```
sg_items(id, raw = FALSE, verbose = FALSE)
```

Arguments

id	character, ID of the collection for which to retrieve the items (see sg_collections()).
raw	logical, defaults to FALSE (see Return).
verbose	logical, defaults to FALSE. If set TRUE the some messages are printed.

Value

If raw = TRUE a list of lists is returned with the raw (json decoded) result from the API calls. Each element in the list corresponds to one API call (paging).

Else (default) the function tries to auto-detect the format. As an example, if the returned data are of type "FeatureCollection"s The data is converted to a simple feature data frame which is then returned. If now only "FeatureCollection" is implemented, if the returned data is of a different type the raw list is returned (TODO).

Author(s)

Reto

Examples

```
## Not run:
## Retrieving items for a collection w/ features
CID <- "ch.meteoschweiz.ogd-smn"
items <- sg_items(CID)

## Visualizing the features
library("ggplot2")
ggplot(st_transform(items, crs = st_crs(2056))) +
    geom_sf(col = "lightgreen") + geom_sf_text(aes(label = id))

## End(Not run)
```

sg_options *Setting* swissgeo *Options*

Description

Allows the user to overwrite a series of default options used in the package, mainly fine-tuning how the HTTPS requests are handled.

Usage

```
sg_options(
  timeout = 20,
  retry = 3,
  capacity = 3,
  filltime = 3,
  apiurl = NULL,
  verbose = FALSE
)
```

Details

`Defaults to 20 seconds.`

A series of default options is set every time the package is attached. They can be overruled by the user calling this function.

Value

Invisibly returns the current swissgeo settings, these are used in the `sg_*` functions for request handling.

Author(s)

Reto

sg_send_api_request *Get Request Helper Function*

Description

Auxilary function to send HTTPS requests with additional error handling. Requests are sent over the httr2 requests object (see `sg_options()` for details).

Usage

```
sg_send_api_request(
  req,
  query = NULL,
  paging = FALSE,
  limit = 100L,
  verbose = FALSE,
  ...
)
```

Arguments

query	NULL (default) or a list with get parameters to be sent with the request. Warning: If the URL contains get requests and query is not NULL, the parameters in the URL (url) will be overwritten!
paging	logical, defaults to FALSE. If TRUE we will send an initial request is sent to the url which returns a maximum of 100 entries. It is then checked if a 'next' link is included in the response used to request the next batch of up to 100 entries until finished.
limit	integer, defaults to 100L. Number of entries to be requested if paging = TRUE.
verbose	logical, defaults to FALSE. If set TRUE the some messages are printed.
...	forwarded to the httr::GET() function. TODO(R)

Details

We are expecting a JSON response from the API, thus we expect that the request content is decoded to a list by the httr package. IF paging = FALSE this list is returned. If paging = TRUE a list of lists is returned, where each element in the main list contains the results of one API call following the 'next' links (see argument paging).

If we do not get a proper HTTP status code, or the extracted content is not a list, this function will throw an error.

Value

List with the decoded information.

Author(s)

Reto

Index