Package 'gdldata'

September 20, 2023

Title 'Global Data Lab' R API

Version 0.1

Description Retrieve datasets from the 'Global Data Lab' website <https://globaldatalab.org> directly into R data frames. Functions are provided to reference available options (indicators, levels, countries, regions) as well.

Depends R (>= 3.4)

Imports httr2, methods

Suggests magrittr

URL https://docs.globaldatalab.org/gdldata/,
https://github.com/GlobalDataLab/R-data-api

BugReports https://github.com/GlobalDataLab/R-data-api/issues

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Global Data Lab [cph],
Aaron van Geffen [aut, cre]

Maintainer Aaron van Geffen <aaron.vangeffen@ru.nl>

Repository CRAN

Date/Publication 2023-09-20 18:40:05 UTC

R topics documented:

`GDLSession-class` *GDLSession class*

Description

GDLSession class

`gdl_countries` *Get country list*

Description

Returns a list of countries available in the current dataset.

Usage

```
gdl_countries(session)
```

Arguments

`session`	A valid GDL session object to interface with.

Value

A data frame containing a list of countries for the dataset.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
# Request list of countries
countries <- gdl_countries(session)
head(countries, n=10)

## End(Not run)
```

`gdl_indicators` *Get indicator list*

Description

Returns the list of indicators available in the current dataset.

Usage

```
gdl_indicators(session)
```

Arguments

`session`	A valid GDL session object to interface with.

Value

A data frame containing a list of indicators for the dataset.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
# Request list of available indicators
indicators <- gdl_indicators(session)
head(indicators, n=10)

## End(Not run)
```

gdl_levels *Get level list*

Description

Returns a list of data levels available in the current dataset.

Usage

```
gdl_levels(session)
```

Arguments

session	A valid GDL session object to interface with.

Value

A data frame containing a list of levels for the dataset.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
# Request list of available levels
levels <- gdl_levels(session)
head(levels, n=10)

## End(Not run)
```

gdl_regions *Get region list*

Description

Returns a list of regions available for a particular country.

Usage

```
gdl_regions(session, country)
```

Arguments

session	A valid GDL session object to interface with.
country	An ISO3 country code.

Value

A data frame containing a list of regions for the country.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
# Request list of regions for India
regions <- gdl_regions(session, 'IND')
head(regions, n=10)

## End(Not run)
```

gdl_request *Data request function*

Description

Data request function

Usage

```
gdl_request(session)
```

Arguments

session	A valid GDL session object to interface with.

Value

A data frame containing the data returned from the GDL API.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
# Customize parameters
session <- set_indicator(session, 'iwi')
session <- set_country(session, 'IND')
# Finally, request the data from GDL
iwi_india <- gdl_request(session)
iwi_india[1:5, 3:8]
# (showing only the five rows and columns for illustrative purposes)

## End(Not run)
```

`gdl_session` *GDL session constructor*

Description

Returns a new GDL session object

Usage

```
gdl_session(token)
```

Arguments

`token`	A valid GDL API token, obtainable from GlobalDataLab.org

Value

A GDL session object for the token with default indicators set.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))

## End(Not run)
```

`set_countries` *Set countries to retrieve data for*

Description

The countries to retrieve GDL indicator data for.

Usage

```
set_countries(session, countries)
```

Arguments

`session`	A valid GDL session object to interface with.
`countries`	A vector of ISO3 country codes.

Value

An amended GDL session object.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
# Customize parameters
session <- set_dataset(session, 'shdi')
session <- set_countries(session, c('BEL', 'LUX', 'NLD'))

## End(Not run)
```

`set_countries_all` *Set session to retrieve data for all available countries*

Description

Switch the session to retrieve data for all available countries, rather than a specific set of countries.

Usage

```
set_countries_all(session)
```

Arguments

`session`	A valid GDL session object to interface with.

Value

An amended GDL session object.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
session <- set_countries_all(session)

## End(Not run)
```

set_country *Set country to retrieve data for*

Description

The country to retrieve GDL indicator data for.

Usage

```
set_country(session, country)
```

Arguments

`session` A valid GDL session object to interface with.

`country` An ISO3 country code.

Value

An amended GDL session object.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
session <- set_country(session, 'IND')

## End(Not run)
```

set_dataset *Set session to retrieve data from a particular dataset*

Description

Switch the session to retrieve data from the dataset specified.

Usage

```
set_dataset(session, dataset)
```

Arguments

`session` A valid GDL session object to interface with.

`dataset` Dataset identifier (string)

Value

An amended GDL session object.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
session <- set_dataset(session, 'shdi')

## End(Not run)
```

---

set_extrapolation_years_linear

Set the number of years to extrapolate linearly.

Description

Sets the number of years to linearly extrapolate at dataset edges. Turns extrapolation on if it isn't already. Overrides the number of nearest years if set.

Usage

```
set_extrapolation_years_linear(session, years)
```

Arguments

`session`	A valid GDL session object to interface with.
`years`	Number of years to extrapolate (integer)

Value

An amended GDL session object.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
session <- set_extrapolation_years_linear(session, 3)

## End(Not run)
```

set_extrapolation_years_nearest *Set the number of years to fill out using nearest available data*

Description

Sets the number of years to fill out using nearest available data at dataset edges, as a means of extrapolation. Turns extrapolation on if it isn't already. Overrides the number of years to linearly extrapolate if set.

Usage

```
set_extrapolation_years_nearest(session, years)
```

Arguments

session	A valid GDL session object to interface with.
years	Number of years to copy (integer)

Value

An amended GDL session object.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
session <- set_extrapolation_years_nearest(session, 3)

## End(Not run)
```

set_indicator *Set the indicator to retrieve*

Description

Sets the indicator to retrieve from the dataset.

Usage

```
set_indicator(session, indicator)
```

Arguments

session	A valid GDL session object to interface with.
indicator	Indicator to retrieve (string)

Value

An amended GDL session object.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
session <- set_indicator(session, 'iwi')

## End(Not run)
```

---

set_indicators *Set the indicators to retrieve*

Description

Sets the indicators to retrieve from the dataset.

Usage

```
set_indicators(session, indicators)
```

Arguments

session	A valid GDL session object to interface with.
indicators	Vector of indicators to retrieve (string)

Value

An amended GDL session object.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
session <- set_indicators(session, c('fridge', 'cellphone', 'tv'))

## End(Not run)
```

set_interpolation *Set interpolation state*

Description

Turns interpolation on (T) or off (F).

Usage

```
set_interpolation(session, state)
```

Arguments

`session`	A valid GDL session object to interface with.
`state`	Whether or not to use interpolation (boolean)

Value

An amended GDL session object.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
session <- set_interpolation(session, TRUE)

## End(Not run)
```

set_levels *Set data levels to retrieve data for*

Description

Specify which data levels to retrieve data for. A list of levels may be obtained through `gdl_levels`.

Usage

```
set_levels(session, levels)
```

Arguments

`session`	A valid GDL session object to interface with.
`levels`	Vector of level identifiers (integers)

Value

An amended GDL session object.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
session <- set_levels(session, c(1,4))

## End(Not run)
```

---

set_year *Set year to retrieve data for*

Description

Specify which year to retrieve data for. Only used when retrieving multiple indicators at once.

Usage

```
set_year(session, year)
```

Arguments

session	A valid GDL session object to interface with.
year	Year to retrieve data for (integer)

Value

An amended GDL session object.

Examples

```
## Not run:
# Create a session using your API token (provided by environment here)
session <- gdl_session(Sys.getenv('GDL_API_TOKEN'))
session <- set_year(session, 2021)

## End(Not run)
```

`show.GDLSession` *GDLSession show function*

Description

This is a user-friendly show function for the GDLSession class, hiding internals from simple print statements.

Usage

```
show.GDLSession
```

Format

An object of class `character` of length 1.

Index