Package ‘gie’

June 3, 2024

Title  API Wrapper for the Natural Gas Transparency Platforms of Gas Infrastructure Europe

Version  0.1.2

Description  Providing access to the API for Gas Infrastructure Europe's natural gas transparency platforms <https://agsi.gie.eu/> and <https://alsi.gie.eu/>. Lets the user easily download metadata on companies and gas storage units covered by the API as well as the respective data on regional, country, company or facility level.

License  MIT + file LICENSE

Imports  curl, dplyr, httr, lubridate, magrittr, purrr, stringr

Suggests  rvest, testthat, textutils, vcr (>= 0.6.0)

Encoding  UTF-8

RoxygenNote  7.3.0

Depends  R (>= 2.10)

LazyData  true

NeedsCompilation  no

Author  Yannik Buhl [aut, cre]

Maintainer  Yannik Buhl <ybuhl@posteo.de>

Repository  CRAN

Date/Publication  2024-06-02 23:20:08 UTC

R topics documented:

  check_giedata2input .................................................. 2
  check_giedatainput .................................................. 3
  check_gielistinginput ............................................... 4
  check_gieunavinput .................................................. 4
  construct_url ....................................................... 5
  countryinfo ......................................................... 5
  extract_listelements ............................................... 6
check_giedata2input

Description

check_giedata2input

Usage

check_giedata2input(
    countries,
    companies,
    facilities,
    from,
    to,
    date,
    size,
    timeout,
    database,
    verbose,
    apikey
)

Arguments

countries Passed from data function for check
companies Passed from data function for check
facilities Passed from data function for check
from Passed from data function for check
to Passed from data function for check
check_giedatainput

Arguments

country
company
facility
from
to
date
size
timeout
database
verbose
apikey

Passed from data function for check
check_gielistingleinput

Description

check_gielistingleinput

Usage

check_gielistingleinput(region, country, facilities, database, apikey)

Arguments

region  Passed from listing function for check
country  Passed from listing function for check
facilities  Passed from listing function for check
database  Passed from listing function for check
apikey  Passed from listing function for check

check_gieunavinput

Description

check_gieunavinput

Usage

check_gieunavinput(country, start, end, type, end_flag, timeout, size, database, apikey)
**construct_url**

**Arguments**

- **country**: Passed from data function for check
- **start**: Passed from data function for check
- **end**: Passed from data function for check
- **type**: Passed from data function for check
- **end_flag**: Passed from data function for check
- **timeout**: Passed from data function for check
- **size**: Passed from data function for check
- **database**: Passed from data function for check
- **apikey**: Passed from data function for check

**construct_url**

---

**Description**

**construct_url**

**Usage**

**construct_url**(url, query)

**Arguments**

- **url**: Base URL
- **query**: Query list

---

**countryinfo**

*Mapping of country name and two-character country code used by GIE*

---

**Description**

A dataset to help you switch between country name and country code in the usage of 'gie', as gie_listing() and gie_load() work with different modes of the country (name vs. two-character code).

**Usage**

**countryinfo**
Format

countryinfo:
A data frame with 17 rows and 2 columns:

name Country name
code Two-character country code

Source

https://www.gie.eu/

extract_listelements

Description

extract_listelements

Usage

extract_listelements(listelement)

Arguments

listelement List element to extract from

getrequest_general

Description

getrequest_general

Usage

getrequest_general(
database,
target,
page,
size,
timeout,
pages = NULL,
apikey,
verbose = FALSE,
...)
)
gie_batchload

**Arguments**

- **database**
  Database name
- **target**
  The API endpoint to target
- **page**
  The page to retrieve
- **size**
  The page size of the request
- **timeout**
  Seconds to delay the batch request
- **pages**
  Number of total pages of existing request
- **apikey**
  API key
- **verbose**
  Enable verbose mode
- ... Further valid API parameters

**Value**

Raw results

---

**gie_batchload**

*gie_batchload – Load data in batch*

**Description**

Function to download data from GIE’s AGSI+/ALSI+ API in bulk

**Usage**

```r
gie_batchload(
  countries, 
  companies = NULL, 
  facilities = NULL, 
  from = NULL, 
  to = NULL, 
  date = NULL, 
  size = 30, 
  timeout = 3, 
  database = "agsi", 
  verbose = FALSE, 
  apikey = Sys.getenv("GIE_APIKEY")
)
```

**Arguments**

- **countries**
  Character. Specify the countries of interest as two-digit country codes (e.g., 'DE', 'IE').
  Must be of length one (i.e., one country) if you want to specify multiple companies and/or multiple facilities.
gie_getnews

Description
A function that conveniently outputs all 'News' items published by GIE with regards to AGSI+ or ALSI+ platforms

Usage
gie_getnews(database, html_parsed = FALSE, apikey = Sys.getenv("GIE_APIKEY"))
gie_listing

**Arguments**

database  Character. The data base for which the 'News' items should be returned ('agsi' or 'alsi').

html_parsed Logical. Some of the columns in the resulting data.frame might contain HTML tags and other encodings. If set to 'TRUE', this parameter will result in the respective column values being decoded so there are no HTML residuals left. This requires the 'rvest' package to be installed and loaded. Defaults to 'FALSE'.

apikey  Character. Your personal API key.

**Value**

A data.frame with all the news for the respective data base.

**Examples**

```r
# Not run:
news <- gie_getnews(database = "alsi", html_parsed = TRUE)
# End(Not run)
```

---

gie_listing – Load info on companies and facilities

**Description**

Function to download raw or parsed results for the countries, companies and facilities available from the AGSI+/ALSI+ API of GIE. The EIC codes of the results can be used in turn to download the actual data using gie_load().

**Usage**

```r
gie_listing(
  region = NULL,
  country = NULL,
  facilities = FALSE,
  database = "agsi",
  apikey = Sys.getenv("GIE_APIKEY")
)
```
Arguments

region Character. The broader region you want results for (can be 'Europe' or 'Non-EU').
country Character. The country you want the results for (must be the written-out name (e.g., "Germany"), NOT the two-digit country code). If you use this parameter, you have to specify the 'region' parameter accordingly.
facilities Logical. If TRUE, facility data will be added to the country or company results. If you use this parameter, 'region' and 'country' have to be set. Defaults to FALSE.
database Character. The type of API you want to address (’agsi’ or ’alsi’).
apikey Character. Your personal API key.

Value

Data.frame with results

Examples

```r
## Not run:
gie_listing(region = "Europe", country = "Germany", facilities = TRUE)
## End(Not run)
```

Description

gie_listinghierarchy

Usage

gie_listinghierarchy(raw_results, region, country, facilities, database)

Arguments

raw_results Raw results from API call
region Region to filter for
country Country to filter for
facilities Should facilities be exported as well
database Database to get info from
**Description**

Function to download data from GIE’s AGSI+ and ALSI+ API

**Usage**

```r
gie_load(
  country,
  company = NULL,
  facility = NULL,
  from = NULL,
  to = NULL,
  date = NULL,
  size = 30,
  timeout = 3,
  database = "agsi",
  verbose = FALSE,
  apikey = Sys.getenv("GIE_APIKEY")
)
```

**Arguments**

- **country** Character. Specify the country of interest as two-digit country code (e.g., 'DE', 'IE').
- **company** Character. EIC code for the requested company.
- **facility** Character. EIC code for the requested facility.
- **from** Character. Specify the start of the time span you are interested in downloading (format: YYYY-MM-DD).
- **to** Character. Specify the end of the time span you are interested in downloading (format: YYYY-MM-DD).
- **date** Character. If you want to have data only for one date. If you set 'date', you cannot set the 'from' and/or 'to' parameters (format: YYYY-MM-DD).
- **size** Integer. The number of results per page.
- **timeout** Numeric. If the amount of pages of your request exceeds 60, a timeout will be enforced to prevent the API from timing out. Defaults to 3 seconds, any values must be set in seconds, too.
- **database** Character. The type of API you want to address ('agsi' or 'alsi').
- **verbose** Logical. Prints information on function progress to the console (default: FALSE).
- **apikey** Character. Your personal API key.
Value

A data.frame or list with the results.

Examples

## Not run:
gie_load(country = "DE", date = "2022-01-03", database = "alsi")
## End(Not run)

gie_unav

gie_unav – Download info on unavailabilities

Description

Function to download data on AGSI+ and ALSI+ unavailability

Usage

gie_unav(
country = NULL,
start = NULL,
end = NULL,
type = NULL,
end_flag = NULL,
timeout = 3,
size = 30,
database = "agsi",
apikey = Sys.getenv("GIE_APIKEY")
)

Arguments

country Character. Specify the country of interest as two-digit country code (e.g., 'DE', 'IE').
start Character. Specify the start of the time span you are interested in downloading (format: YYYY-MM-DD).
end Character. Specify the end of the time span you are interested in downloading (format: YYYY-MM-DD).
type Character. The type of unavailability info you want (choose between 'planned' and 'unplanned').
end_flag Character. The end flag for the unavailability info (choose between 'confirmed' and 'estimate').
timeout  Numeric. If the amount of pages of your request exceeds 60, a timeout will be enforced to prevent the API from timing out. Defaults to 3 seconds, any values must be set in seconds, too.
size  Integer. The number of results per page.
database  Character. The type of API you want to address ('agsi' or 'alsi').
apikey  Character. Your personal API key.

Value
A data.frame or list with the results.

Examples
```r
## Not run:
gie_unav(country = "DE", date = "2022-01-03", database = "alsi")
## End(Not run)
```

---

is_response

Description
is_response

Usage
is_response(x)

Arguments
x  Object to check

---

parseresult

Description
parseresult

Usage
parseresult(raw_results, database)

Arguments
raw_results  Raw results from GET request
database  The database to be used
Description

A function to parse unavailability results

Usage

parse_unav(raw_results)

Arguments

raw_results Raw results

Value

A data.frame

Description

send_getrequest

Usage

send_getrequest(url, apikey, ...)

Arguments

url URL for GET request
apikey API key
... Further API parameters
**setnull**

Description

setnull

Usage

setnull(data, x)

Arguments

data | Data.frame or list element  
x | Name of element to delete

---

**strip_html**

Description

A function to HTML decode a character vector of length > 1

Usage

strip_html(string)

Arguments

string | String to strip HTML tags from

Value

A vector with HTML decoded text
try_GET

Description

try_GET

Usage

try_GET(url, apikey, ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>URL for API call</td>
</tr>
<tr>
<td>apikey</td>
<td>API key</td>
</tr>
<tr>
<td>...</td>
<td>Further parameter for http::GET</td>
</tr>
</tbody>
</table>
Index

* datasets
  countryinfo, 5
check_giedata2input, 2
check_giedatainput, 3
check_gielistinginput, 4
check_gieunavinput, 4
construct_url, 5
countryinfo, 5
extract_listelements, 6
getrequest_general, 6
gie_batchload, 7
gie_getnews, 8
gie_listing, 9
gie_listinhierarchy, 10
gie_load, 11
gie_unav, 12
is_response, 13
parse_unav, 14
parseresult, 13
send_getrequest, 14
setnull, 15
strip_html, 15
try_GET, 16