Package ‘gie’

February 28, 2024

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

Version 0.1.1

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 2024-02-28 18:40:02 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
Description

`check_giedatainput`

Usage

```python
check_giedatainput(
    country,  # Passed from data function for check
    company,  # Passed from data function for check
    facility,  # Passed from data function for check
    from,      # Passed from data function for check
    to,        # Passed from data function for check
    date,      # Passed from data function for check
    size,      # Passed from data function for check
    timeout,   # Passed from data function for check
    database,  # Passed from data function for check
    verbose,   # Passed from data function for check
    apikey     # Passed from data function for check
)
```

Arguments

country  # Passed from data function for check
company  # Passed from data function for check
facility # Passed from data function for check
from     # Passed from data function for check
to       # Passed from data function for check
date     # Passed from data function for check
size     # Passed from data function for check
timeout  # Passed from data function for check
database # Passed from data function for check
verbose  # Passed from data function for check
apikey   # Passed from data function for check
check_gieunavinput

Description

check_gieunavinput

Usage

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>region</td>
<td>Passed from listing function for check</td>
</tr>
<tr>
<td>country</td>
<td>Passed from listing function for check</td>
</tr>
<tr>
<td>facilities</td>
<td>Passed from listing function for check</td>
</tr>
<tr>
<td>database</td>
<td>Passed from listing function for check</td>
</tr>
<tr>
<td>apikey</td>
<td>Passed from listing function for check</td>
</tr>
</tbody>
</table>

check_gieunavinput  

Description

check_gieunavinput

Usage

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>region</td>
<td>Passed from listing function for check</td>
</tr>
<tr>
<td>country</td>
<td>Passed from listing function for check</td>
</tr>
<tr>
<td>facilities</td>
<td>Passed from listing function for check</td>
</tr>
<tr>
<td>database</td>
<td>Passed from listing function for check</td>
</tr>
<tr>
<td>apikey</td>
<td>Passed from listing function for check</td>
</tr>
</tbody>
</table>
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

---

**Description**

**construct_url**

**Usage**

```r
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**

```r
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/

**Usage**

```r
database, target, page, size, timeout, pages = NULL, apikey, verbose = FALSE, ...
```

**Arguments**

- **listelement** List element to extract from
### gie_batchload

#### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>database</td>
<td>Database name</td>
</tr>
<tr>
<td>target</td>
<td>The API endpoint to target</td>
</tr>
<tr>
<td>page</td>
<td>The page to retrieve</td>
</tr>
<tr>
<td>size</td>
<td>The page size of the request</td>
</tr>
<tr>
<td>timeout</td>
<td>Seconds to delay the batch request</td>
</tr>
<tr>
<td>pages</td>
<td>Number of total pages of existing request</td>
</tr>
<tr>
<td>apikey</td>
<td>API key</td>
</tr>
<tr>
<td>verbose</td>
<td>Enable verbose mode</td>
</tr>
<tr>
<td>...</td>
<td>Further valid API parameters</td>
</tr>
</tbody>
</table>

#### 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>countries</td>
<td>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.</td>
</tr>
</tbody>
</table>
companies  A character vector of company EIC codes to get data from. Must be of length one (i.e., one company) if you want to specify multiple facilities.

facilities  A character vector of facility EIC codes to get data from.

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 with results

Examples

```r
## Not run:
gie_batchload(countries = c("DE", "AT", "FR"), date = "2022-04-01")

## End(Not run)
```

gie_getnews  
gie_getnews – Get AGSI+/ALSI+ news

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

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

**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")
)
```

**Value**

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

**Examples**

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

Value

Data.frame with results

Examples

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

## End(Not run)
gie_load – Main download function

Description

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

Usage

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

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

---

### gie_unav – Download info on unavailabilities

#### Description

Function to download data on AGSI+ and ALSI+ unavailability

#### Usage

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

---

is_response is_response

Description
is_response

Usage
is_response(x)

Arguments
x Object to check

parseresult parseresult

Description
parseresult

Usage
parseresult(raw_results, database)

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

**Description**
A function to parse unavailability results

**Usage**
`parse_unav(raw_results)`

**Arguments**
- raw_results: Raw results

**Value**
A data.frame

---

**send_getrequest**

**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_listinghierarchy, 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