Package 'gie'

May 8, 2026

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

Version 0.1.3

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 7.3.2

Depends R (>= 2.10)

LazyData true

NeedsCompilation no

Author Yannik Buhl [aut, cre]

Maintainer Yannik Buhl <ybuhl@posteo.de>

Repository CRAN

Date/Publication 2025-03-29 20:00:02 UTC

Contents

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_getumm 9
gie_listing 10
gie_listinghierarchy 11
gie_load 12
gie_unav 13
is_response 14
is_valid_date 15
parseresult 15
parse_unav 16
send_getrequest 16
setnull 17
strip_html 17
try_GET 18

Index **19**

`check_giedata2input` *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
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_giedatainput` *check_giedatainput*

Description

check_giedatainput

Usage

```
check_giedatainput(
  country,
  company,
  facility,
  from,
  to,
  date,
  size,
  timeout,
  database,
  verbose,
  apikey
)
```

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_gielistinginput *check_gielistinginput*

Description

check_gielistinginput

Usage

```
check_gielistinginput(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 *check_gieunavinput*

Description

check_gieunavinput

Usage

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

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 *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 *extract_listelements*

Description

extract_listelements

Usage

```
extract_listelements(listelement)
```

Arguments

listelement	List element to extract from

---

getrequest_general *getrequest_general*

Description

getrequest_general

Usage

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

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

```
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.

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

```
## 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, html_parsed = FALSE, apikey = Sys.getenv("GIE_APIKEY"))
```

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

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

## End(Not run)
```

---

gie_getumm *gie_getumm – Download info on IIP urgent market messages*

Description

Function to download urgent market messages (UMM) from the Inside Information Platform (IIP)

Usage

```
gie_getumm(
  from = NULL,
  to = NULL,
  published_date = NULL,
  eic_entity = NULL,
  eic_participant = NULL,
  eic_asset = NULL,
  status = NULL,
  direction = NULL,
  timeout = 3,
  size = 300,
  apikey = Sys.getenv("GIE_APIKEY")
)
```

Arguments

from	Character. Date the UMM is being in place from (format: YYYY-MM-DD).
to	Character. Date the UMM is being in place to (format: YYYY-MM-DD).
published_date	Character. Filter by publication date of UMM. Can be partial (e.g.: "2025-02-25", "2025-02", "2025").
eic_entity	Character. EIC code of the UMM issuing entity.
eic_participant	Character. EIC code of the UMM issuing market participant.
eic_asset	Character. EIC code of the asset the UMM is issued for.
status	Character. Can be 'active', 'inactive' or 'dismissed'. Default is all.
direction	Character. Direction of gas flows. Can be "entry" or "exit". Default is all.
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.
apikey	Character. Your personal API key.

Value

A nested list with UMMs for your specific API query to IIP. Note: The results are not parsed for there are too many data variants a user could want.

Examples

```
## Not run:
umm <- gie_getumm(from = "2024-01-01", to = "2024-12-31")

## End(Not run)
```

---

gie_listing *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

```
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

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

## End(Not run)
```

---

gie_listinghierarchy *gie_listinghierarchy*

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 facilties be exported as well
database	Database to get info from

---

gie_load *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

```
## 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

```
## 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

is_valid_date *Check if a string is a valid date*

Description

Check if a string is a valid date

Usage

```
is_valid_date(date_string)
```

Arguments

date_string	Date string to check

Value

TRUE or FALSE

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 *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 *send_getrequest*

Description

send_getrequest

Usage

```
send_getrequest(url, apikey, ...)
```

Arguments

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

setnull *setnull*

Description

setnull

Usage

```
setnull(data, x)
```

Arguments

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

strip_html *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 *try_GET*

Description

try_GET

Usage

```
try_GET(url, apikey, ...)
```

Arguments

url	URL for API call
apikey	API key
...	Further parameter for httr::GET

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_getumm, 9
gie_listing, 10
gie_listinghierarchy, 11
gie_load, 12
gie_unav, 13

is_response, 14
is_valid_date, 15

parse_unav, 16
parseresult, 15

send_getrequest, 16
setnull, 17
strip_html, 17

try_GET, 18