

# Package ‘giedata’

November 26, 2022

**Title** API Wrapper for GIE's AGSI+/ALSI+ Natural Gas Storage  
Transparency Platform

**Version** 0.1.1

**Description** Functions to access the API for Gas Infrastructure Europe's (GIE) AGSI+/ALSI+ natural gas storage transparency platform (<<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

**Encoding** UTF-8

**RoxygenNote** 7.2.0

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

**Suggests** vcr (>= 0.6.0), testthat

**NeedsCompilation** no

**Author** Yannik Buhl [aut, cre]

**Maintainer** Yannik Buhl <ybuhl@posteo.de>

**Repository** CRAN

**Date/Publication** 2022-11-26 21:30:02 UTC

## R topics documented:

check_giedata2input . . . . .	2
check_giedatainput . . . . .	3
check_gielistinginput . . . . .	4
construct_url . . . . .	4
extract_listelements . . . . .	5
getrequest . . . . .	5
get_giedata . . . . .	6
get_giedata2 . . . . .	7
get_gielisting . . . . .	9
get_listinghierarchy . . . . .	10
parseresult . . . . .	10

send_getrequest . . . . .	11
setnull . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

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

---

construct\_url *construct\_url*

---

**Description**

construct\_url

**Usage**

construct\_url(url, query)

**Arguments**

url	Base URL
query	Query list

---

`extract_listelements`    *extract\_listelements*

---

**Description**

`extract_listelements`

**Usage**

`extract_listelements(listelement)`

**Arguments**

`listelement`    List element to extract from

---

`getrequest`            *getrequest*

---

**Description**

`getrequest`

**Usage**

```
getrequest(  
    country,  
    company,  
    facility,  
    from,  
    to,  
    page,  
    date,  
    size,  
    timeout,  
    database,  
    pages = NULL,  
    verbose,  
    apikey  
)
```

**Arguments**

country	Country to get data for
company	Company EIC code
facility	Facility EIC code
from	Date for extraction start
to	Date for extraction end
page	Page number if multiple pages available
date	Date for extraction
size	Page size
timeout	Seconds to timeout if query is large
database	Database name
pages	Total number of pages
verbose	Verbose mode
apikey	API key

---

`get_giedata``get_giedata`

---

**Description**

Function to download data from GIE's AGSI+ API

**Usage**

```
get_giedata(  
  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:
get_giedata(country = "DE", date = "2022-01-03")

## End(Not run)
```

---

```
get_giedata2
```

```
get_giedata2
```

---

**Description**

Function to download data from GIE's AGSI+ API in bulk

**Usage**

```

get_giedata2(
  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**

<code>countries</code>	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.
<code>companies</code>	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.
<code>facilities</code>	A character vector of facility EIC codes to get data from.
<code>from</code>	Character. Specify the start of the time span you are interested in downloading (format: YYYY-MM-DD).
<code>to</code>	Character. Specify the end of the time span you are interested in downloading (format: YYYY-MM-DD).
<code>date</code>	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).
<code>size</code>	Integer. The number of results per page.
<code>timeout</code>	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.
<code>database</code>	Character. The type of API you want to address ('agsi' or 'alsi').
<code>verbose</code>	Logical. Prints information on function progress to the console (default: FALSE).
<code>apikey</code>	Character. Your personal API key.

**Value**

A data.frame with results



**Examples**

```
## Not run:
get_giedata2(countries = c("DE", "AT", "FR"), date = "2022-04-01")

## End(Not run)
```

---

```
get_gielisting      get_gielisting
```

---

**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 to in turn download the actual data using `get_giedata()`.

**Usage**

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

**Arguments**

<code>region</code>	Character. The broader region you want results for (can be 'Europe' or 'Non-EU').
<code>country</code>	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.
<code>facilities</code>	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.
<code>database</code>	Character. The type of API you want to address ('agsi' or 'alsi'). As of yet, only 'agsi' works, 'alsi' support is expected at a later time.
<code>apikey</code>	Character. Your personal API key.

**Value**

Data.frame with results

**Examples**

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

---

get\_listinghierarchy    *get\_listinghierarchy*

---

**Description**

get\_listinghierarchy

**Usage**

```
get_listinghierarchy(raw_results, region, country, facilities)
```

**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

---

parseresult            *parseresult*

---

**Description**

parseresult

**Usage**

```
parseresult(raw_results)
```

**Arguments**

raw_results	Raw results from GET request
-------------	------------------------------

---

send_getrequest	<i>send_getrequest</i>
-----------------	------------------------

---

**Description**

send\_getrequest

**Usage**

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

**Arguments**

url	URL for GET request
apikey	API key

---

setnull	<i>setnull</i>
---------	----------------

---

**Description**

setnull

**Usage**

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

**Arguments**

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

# Index

[check\\_giedata2input](#), [2](#)  
[check\\_giedatainput](#), [3](#)  
[check\\_gielistinginput](#), [4](#)  
[construct\\_url](#), [4](#)

[extract\\_listelements](#), [5](#)

[get\\_giedata](#), [6](#)  
[get\\_giedata2](#), [7](#)  
[get\\_gielisting](#), [9](#)  
[get\\_listinghierarchy](#), [10](#)  
[getrequest](#), [5](#)

[parseresult](#), [10](#)

[send\\_getrequest](#), [11](#)  
[setnull](#), [11](#)