Package 'ecos'

February 8, 2024

Title Economic Statistics System of the Bank of Korea

Version 0.1.6

Description API wrapper to download statistical information from the Economic Statistics System (ECOS) of the Bank of Korea <https://ecos.bok.or.kr/api/#/>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports httr (>= 1.4.3), jsonlite (>= 1.7.2), XML (>= 3.99), stringr (>= 1.4.0)

RoxygenNote 7.2.3

NeedsCompilation no

Author Seokhoon Joo [aut, cre], Jaehyun Joo [ctb]

Maintainer Seokhoon Joo <seokhoonj@gmail.com>

Repository CRAN

Date/Publication 2024-02-08 08:40:02 UTC

R topics documented:

ecos-package *ecos: Economic Statistics System of the Bank of Korea*

Description

API wrapper to download statistical information from the Economic Statistics System (ECOS) of the Bank of Korea https://ecos.bok.or.kr/api/#/.

Details

To use this package, you will first need to get your API key from the website https://ecos.bok.or.kr/api/#/AuthKeyApply. Once you have your key, you can save it as an environment variable for the current session using the ecos.setKey function. Alternatively, you can set it permanently by adding the following line to your .Renviron file:

ECOS_API_KEY = PASTE YOUR API KEY

Any functions that require your API key try to retrieve it via `Sys.getenv("ECOS_API_KEY")` (unless API key is explicitly specified as a function argument).

Author(s)

Maintainer: Seokhoon Joo <seokhoonj@gmail.com>

Other contributors:

- Jaehyun Joo [contributor]

calendar *Calendar for the cycle argument*

Description

The ecos open API has been revised on 2022.06.01. A calendar was created to respond to any type of date format according to the cycle argument.

Usage

```
calendar
```

Format

A data frame with 73049 rows and 6 variables:

D daily

SM semi-monthly

M monthly

Q quarterly

S semi-annually

A annually

Examples

```
calendar
```

`ecos.setKey` *Set ECOS API Key*

Description

Save ECOS API key for the current session. To set it permanently, please add the following line to your .Renvrion file:

ECOS_API_KEY = YOUR API KEY

Usage

```
ecos.setKey(api_key)

ecos.printKey()
```

Arguments

`api_key`	A string specifying ECOS API key

Value

No return value, called to set api key

Examples

```
## Set API Key for the current session
ecos.setKey("your_api_key")

## Check API key
ecos.printKey()
```

keyStatList *Top 100 statistical indicators*

Description

Top 100 statistical indicators

Usage

```
keyStatList(format = c("xml", "json"), lang = c("kr", "en"), count = 1000)
```

Arguments

format	A string specifying the file format to process - xml, json
lang	A string specifying the language of result value - kr (Korean), en (English)
count	An integer specifying the number of requests

Details

```
## Example
keyStatList(lang = "en", count = 100)
```

Value

A data.frame object containing queried information

statItemList *Item list of statistics*

Description

Item list of statistics

Usage

```
statItemList(
  stat_code,
  format = c("xml", "json"),
  lang = c("kr", "en"),
  count = 1000
)
```

Arguments

stat_code	A string specifying the statistical table code
format	A string specifying the file format to process - xml, json
lang	A string specifying the language of result value - kr (Korean), en (English)
count	An integer specifying the number of requests

Details

```
## Example
statItemList(lang = "en", count = 100, stat_code = "902Y001")
```

Value

A data.frame object containing queried information

---

statMeta *Retrieve statistical meta DB*

Description

Retrieve statistical meta DB

Usage

```
statMeta(meta, format = c("xml", "json"), lang = c("kr", "en"), count = 1000)
```

Arguments

meta	A string specifying the name of meta DB to query
format	A string specifying the file format to process - xml, json
lang	A string specifying the language of result value - kr (Korean), en (English)
count	An integer specifying the number of requests

Details

```
## Example
statMeta(lang = "en", meta = "Economic Sentiment Index")
```

Value

A data.frame object containing queried information

statSearch *Search conditional statistics*

Description

Search conditional statistics

Usage

```
statSearch(
  stat_code,
  item_code1,
  item_code2 = "?",
  item_code3 = "?",
  item_code4 = "?",
  cycle,
  start_time,
  end_time,
  format = c("xml", "json"),
  lang = c("kr", "en"),
  count
)
```

Arguments

stat_code	A string specifying the statistical table code
item_code1	A string specifying the statistical item 1 code
item_code2	A string specifying the statistical item 2 code
item_code3	A string specifying the statistical item 3 code
item_code4	A string specifying the statistical item 4 code
cycle	A string specifying the cycle (Annual: A, Semi-Annual: S, Quarterly: Q, Monthly: M, Semi-Monthly: SM, Daily: D)
start_time	A string specifying the start date (according to cycle format: 2015, 2015S1, 2015Q1, 201501, 201501S1, 20150101, etc.)
end_time	A string specifying the end date (according to cycle format: 2015, 2015S1, 2015Q1, 201501, 201501S1, 20150101, etc.)
format	A string specifying the file format to process - xml, json
lang	A string specifying the language of result value - kr (Korean), en (English)
count	An integer specifying the number of requests

Details

```
## Example
statSearch(lang = "en", stat_code = "102Y004", item_code1 = "ABA1", cycle =
"M", start_time = "196001", end_time = "201812")
```

Value

A data.frame object containing queried information

`statTableList` *Table list of statistical tables*

Description

Table list of statistical tables

Usage

```
statTableList(format = c("xml", "json"), lang = c("kr", "en"), count = 1000)
```

Arguments

`format`	A string specifying the file format to process - xml, json
`lang`	A string specifying the language of result value - kr (Korean), en (English)
`count`	An integer specifying the number of requests

Details

```
## Example
statTableList(lang = "en", count = 100)
```

Value

A data.frame object containing queried information

`statWord` *Glossary of Statistical Terms*

Description

Glossary of Statistical Terms

Usage

```
statWord(word, format = c("xml", "json"), lang = c("kr", "en"), count = 1000)
```

Arguments

`word`	A string specifying the term to search
`format`	A string specifying the file format to process - xml, json
`lang`	A string specifying the language of result value - kr (Korean), en (English)
`count`	An integer specifying the number of requests

Details

```
## Example
statWord(word = "CPI", lang = "en")
```

Value

A data.frame object containing queried information

Index