Package 'eia'

February 21, 2021

Title API Wrapper for 'US Energy Information Administration' Open Data

Version 0.3.7

Description Provides API access to data from the 'US Energy Information Administration' ('EIA') <https://www.eia.gov/>.
Use of the API requires a free API key obtainable at <https://www.eia.gov/opendata/register.php>.
The package includes functions for searching 'EIA' data categories and importing time series and geoset time series datasets.
Datasets returned by these functions are provided in a tidy format or alternatively in more raw form.
It also offers helper functions for working with 'EIA' date strings and time formats and for inspecting different summaries of series metadata.
The package also provides control over API key storage and caching of API request results.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL https://docs.ropensci.org/eia/ (website)
https://github.com/ropensci/eia

BugReports https://github.com/ropensci/eia/issues

Imports tibble, magrittr, httr, jsonlite, dplyr, purrr, memoise, lubridate, readxl

Suggests testthat, knitr, rmarkdown, covr, tidyr, ggplot2

VignetteBuilder knitr

Language en-US

RoxygenNote 7.1.1

NeedsCompilation no

Author Matthew Leonawicz [aut, cre] (<https://orcid.org/0000-0001-9452-2771>)

Maintainer Matthew Leonawicz <mfleonawicz@gmail.com>

Repository CRAN

Date/Publication 2021-02-20 23:00:05 UTC

R topics documented:

`eia` *eia: EIA API wrapper*

Description

This package provides API access to data from the US Energy Information Administration (EIA).

`eiadate` *EIA date parsing*

Description

Helper functions for manipulating and converting between regular year-month-day date strings and EIA date string notation.

Usage

```
eiadate_to_date(x)

date_to_eiadate(x, date_format = c("A", "Q", "M", "W", "D", "H"))

eiadate_to_date_seq(start, end, weekly = FALSE)
```

Arguments

`x`	character, EIA date string; character or date object for regular dates. See details.
`date_format`	EIA date format: "A", "Q", "M", "W", "D", "H". These stand for annual, quarterly, monthly, weekly, daily, hourly. See details.
`start`	start EIA date or date.
`end`	end EIA date or date.
`weekly`	logical. See details.

Details

There is no reason to mix EIA date formats in this context. Functions that take EIA date strings expect a consistent format. Also, EIA date formats are parsed automatically from the dates themselves. However, daily and weekly use the same format. Too avoid ambiguity in `eia_date_seq`, daily is assumed; set `weekly = TRUE` to treat as weekly.

When providing a real date or date string, such as to `date_to_eiadate`, dates should be in `yyyy-mm-dd` format, or at least any format that can be parsed by `lubridate::ymd` or `lubridate::ymd_hms` for dates and hourly date times, respectively.

`"HL"` is not a supported date format. Use `"H"`. The API does not translate the date and time when using `"HL"` anyhow; it simply appends the date string with the number of hours time difference.

Examples

```
eiadate_to_date(c("201803", "201804"))

date_to_eiadate("2018-05-14", "A")
date_to_eiadate("2018-05-14", "Q")
date_to_eiadate("2018-05-14", "M")

(x <- eiadate_to_date_seq("2018Q1", "2018Q4"))
date_to_eiadate(x, "Q")

(x <- eiadate_to_date("20190102T16Z"))
date_to_eiadate(x, "H")
(x <- eiadate_to_date_seq("20190102T16Z", "20190102T19Z"))
date_to_eiadate(x, "H")
```

---

eia_cats *EIA categories*

Description

Obtain EIA categories.

Usage

```
eia_cats(id = NULL, tidy = TRUE, cache = TRUE, key = eia_get_key())

eia_child_cats(id, cache = TRUE, key = eia_get_key())

eia_parent_cats(id, cache = TRUE, key = eia_get_key())
```

Arguments

id	integer, category ID. If `NULL`, the API root category.
tidy	logical, return a tidier result. See details.
cache	logical, cache result for duration of R session using memoization. See details.

key — API key: character if set explicitly; not needed if key is set globally. See eia_set_key.

Details

By default, additional processing is done to return a list containing tibble data frames. Set tidy = FALSE to return only the initial list result of jsonlite::fromJSON. Set tidy = NA to return the original JSON as a character string.

Set to cache = FALSE to force a new API call for updated data. Using FALSE always makes a new API call and returns the result from the server. TRUE uses memoization on a per R session basis, caching the result of the function call in memory for the duration of the R session. You can reset the entire cache by calling eia_clear_cache.

eia_child_cats returns only the immediate child categories. eia_parent_cats returns all parents. These are wrappers around eia_cats and always return a tibble data frame.

Value

for eia_cats, a list of tibble data frames (or a less processed list, or character, depending on tidy value); others functions return a tibble data frame.

Examples

```
## Not run:
# use eia_set_key() to store stored API key
eia_cats()

eia_child_cats(389) # immedate children
eia_parent_cats(742) # all parents

## End(Not run)
```

---

eia_clear_cache *Clear API results cache*

Description

Reset the results of API calls that are currently cached in memory.

Usage

```
eia_clear_cache()

eia_clear_cats()

eia_clear_series()

eia_clear_geoset()
```

Details

eia_clear_cache clears the entire cache. The other functions clear the cache associated with specific endpoints.

Examples

```
## Not run:
key <- Sys.getenv("EIA_KEY") # your stored API key
system.time(eia_cats(key))
system.time(eia_cats(key))
eia_clear_cache()
system.time(eia_cats(key))

## End(Not run)
```

---

eia_geoset *EIA geoset data*

Description

Obtain EIA geoset data.

Usage

```
eia_geoset(
  id,
  region,
  relation = NULL,
  start = NULL,
  end = NULL,
  n = NULL,
  tidy = TRUE,
  cache = TRUE,
  key = eia_get_key()
)
```

Arguments

id	character, geoset series ID, may be a vector. See details.
region	character, region ID, may be a vector. Data available for the intersection of id and region is returned.
relation	logical, make a geoset relation query instead of a geoset query. The series id is the same but is queried differently. Currently not supported, see details.
start	start date. Providing only a start date will return up to the maximum 100 results if available.
end	end date. Providing only an end date will a single result for that date.

n	integer, length of series to return ending at most recent value or at end date if also provided. Ignored if start is not NULL.
tidy	logical, return a tidier result. See details.
cache	logical, cache result for duration of R session using memoization. See details.
key	API key: character if set explicitly; not needed if key is set globally. See eia_set_key.

Details

id may be a vector. This should only be done with tidy = TRUE if the tidied results can be properly row bound. The geoset API calls allow multiple regions, but the API expects a single series ID. This function allows multiple series, but must make one API call per series ID. There is an expectation of similarly formatted series that can be row bound. If the IDs are for differently structured data that cannot be tidily row bound, you may as well make separate requests since each requires a unique API call either way.

By default, additional processing is done to return a tibble data frame. Set tidy = FALSE to return only the initial list result of jsonlite::fromJSON. Set tidy = NA to return the original JSON as a character string.

Set to cache = FALSE to force a new API call for updated data. Using FALSE always makes a new API call and returns the result from the server. TRUE uses memoization on a per R session basis, caching the result of the function call in memory for the duration of the R session. You can reset the entire cache by calling eia_clear_cache().

The EIA relation API endpoint is officially supported according to the online EIA API documentation, but that endpoint does not appear to function at the time of current package release.

Value

a tibble data frame (or a list, or character, depending on tidy value)

See Also

eia_clear_cache

Examples

```
## Not run:
# use eia_set_key() to store stored API key
id <- paste0("ELEC.GEN.ALL-99.", c("A", "Q", "M"))
region <- c("USA-CA", "USA-NY")

eia_geoset(id[1], region[1], start = 2016)
eia_geoset(id[2], region, n = 5)
eia_geoset(id[3], region[2], end = 2016, n = 5)

# multiple series counted as a single API call
x <- eia_geoset(id, region[1], end = 2016, n = 2)
x[, c("region", "data")]

# Use direct US state abbreviations or names;
```

```
# Use US Census region and division names.
x <- eia_geoset(id[2], c("AK", "New England"), end = 2016, n = 1)
x[, c("region", "data")]

## End(Not run)
```

`eia_key` *Set and get API key*

Description

Set and get API key

Usage

```
eia_set_key(key, store = c("env", "options", "sysenv"))

eia_get_key(store = c("env", "options", "sysenv"))
```

Arguments

`key`	character, API key.
`store`	character, method for storing API key. See details.

Details

Setter and getter helpers allow you to store your EIA API key in one of three ways. Their use is optional. You can always pass the API key string to the `key` argument of any package function that requires it, but you do not have to.

By default the `key` argument for these functions is `key = eia_get_key()`. If your key has been stored in a manner that can be retrieved, then you can call all the package API functions without having to provide the `key` argument repeatedly.

Value

`eia_get_key` returns the key string or `NULL` with a warning. `eia_set_key` returns a success message or an error.

Key storage methods

If you have already set your key globally somewhere using `eia_set_key`, `eia_get_key` will retrieve it. You can add the `EIA_KEY = "yourkey"` key-value pair to `options()` or as a system environment variable yourself and `eia_get_key` will pick it up as long as you use the name `EIA_KEY`. For convenience you can do this in your R session with `eia_set_key`. It gives you three options for how to store the key. The default is to use the `eia` package environment that is created when the package is loaded.

Precedence

Choose one method when setting a key. When getting the key, the three locations are checked in the order: package environment, `options()`, then the system environment. To override the order, specify the method explicitly and the check will only occur there. This also makes it possible to override a system level key by working with one stored in the package environment or `options()`.

Persistence

Note that none of these three storage methods, including `"sysenv"` are persistent; the stored key is lost when the R session is terminated. A key that is stored outside of R as a system environment variable is retrievable with `eia_get_key`, just like those set in an R session with `eia_set_key` and `store = "sysenv"`. However, if you truly want the key to persist as an environment variable when R terminates, you must manually add it somewhere like `.Renviron`; `Sys.setenv` in R cannot achieve this.

Examples

```
eia_set_key("fake")
eia_get_key()
# eia_get_key("options") returns an error if not set
```

---

eia_report *Download data for various EIA reports*

Description

These functions download data for various EIA reports found on the EIA website but not necessarily available through the EIA API.

Usage

```
eia_report(id, ...)

report_drilling_productivity()
```

Arguments

`id`	character, the report ID. See details for the list of available reports.
`...`	arguments passed to individual report data functions.

Details

The wrapper function and the individual report functions do not make API calls and do not require an API key.

Value

a list, typically a list of data frames

Examples

```
## Not run:
x <- eia_report("drilling productivity")

## End(Not run)
```

`eia_series` *EIA data series*

Description

Obtain EIA data series.

Usage

```
eia_series(
  id,
  start = NULL,
  end = NULL,
  n = NULL,
  tidy = TRUE,
  cache = TRUE,
  key = eia_get_key()
)
```

Arguments

`id`	character, series ID, may be a vector.
`start`	start date. Providing only a start date will return up to the maximum 100 results if available.
`end`	end date. Providing only an end date will a single result for that date.
`n`	integer, length of series to return ending at most recent value or at `end` date if also provided. Ignored if `start` is not `NULL`.
`tidy`	logical, return a tidier result. See details.
`cache`	logical, cache result for duration of R session using memoization. See details.
`key`	API key: character if set explicitly; not needed if key is set globally. See `eia_set_key`.

Details

By default, additional processing is done to return a tibble data frame. Set `tidy = FALSE` to return only the initial list result of `jsonlite::fromJSON`. Set `tidy = NA` to return the original JSON as a character string.

Set to `cache = FALSE` to force a new API call for updated data. Using `FALSE` always makes a new API call and returns the result from the server. `TRUE` uses memoization on a per R session basis, caching the result of the function call in memory for the duration of the R session. You can reset the entire cache by calling `eia_clear_cache()`.

Value

a tibble data frame (or a list, or character, depending on `tidy` value)

See Also

`eia_clear_cache`

Examples

```
## Not run:
# use eia_set_key() to store stored API key
id <- paste0("ELEC.GEN.ALL-AK-99.", c("A", "Q", "M"))

x1 <- eia_series(id[1], start = 2016)
x2 <- eia_series(id[2], n = 5)
x3 <- eia_series(id[3], end = 2016, n = 5)
x1$data[[1]]
x2$data[[1]]
x3$data[[1]]

# multiple series counted as a single API call
x <- eia_series(id, end = 2016, n = 2)
x$data

## End(Not run)
```

---

eia_series_metadata *EIA series metadata*

Description

Make a small request to obtain a data frame containing metadata.

Usage

```
eia_series_metadata(id, cache = TRUE, key = eia_get_key())

eia_series_updates(id, cache = TRUE, key = eia_get_key())

eia_series_dates(id, cache = TRUE, key = eia_get_key())

eia_series_range(id, cache = TRUE, key = eia_get_key())

eia_series_cats(id, tidy = TRUE, cache = TRUE, key = eia_get_key())
```

Arguments

id	character, series ID, may be a vector.
cache	logical, cache result for duration of R session using memoization.
key	API key: character if set explicitly; not needed if key is set globally. See eia_set_key.
tidy	logical, return a tidier result. See details.

Details

Dates are provided in eia_series_dates for the convenience of working with the EIA date string format; for example: maintaining order, generating sequences, computing intervals, and other operations that work well with dates but would be difficult using arbitrary strings. Keep in mind that of course these are not real dates, in the sense that you cannot map a year to a specific date.

eia_series_updates returns a data frame of most recent series update times for id. Like the other metadata helpers, this does require an API call to the series to obtain the relevant metadata. This can be useful if you are only interested in these update times for a specific set of series IDs. If you need to know the most recent update stamps for a large set of series, you should use eia_updates instead, which makes an API call specifically to the EIA updates endpoint for specific EIA categories by category ID.

eia_series_cats differs from the other functions in that it makes an API call directly to the series categories endpoint. Like other functions that return endpoint-specific output, it accepts the tidy argument for control over output structure. By default, additional processing is done to return a list containing tibble data frames. Set tidy = FALSE to return only the initial list result of jsonlite::fromJSON. Set tidy = NA to return the original JSON as a character string.

Value

a tibble data frame

See Also

eia_updates

Examples

```
## Not run:
# use eia_set_key() to store stored API key
id <- paste0("ELEC.CONS_TOT_BTU.COW-AK-1.", c("A", "Q", "M"))

eia_series_metadata(id)
eia_series_updates(id)
eia_series_dates(id)
eia_series_range(id)
eia_series_cats(id)

## End(Not run)
```

`eia_updates` *EIA data updates*

Description

Obtain information on EIA data series updates for a given category to avoid having to make requests for data that have not been updated since your last request.

Usage

```
eia_updates(
  id = NULL,
  deep = FALSE,
  n = 50,
  start = 1,
  tidy = TRUE,
  key = eia_get_key()
)
```

Arguments

`id`	integer, category ID, may be a vector. If `NULL`, the API root category.
`deep`	logical, if `TRUE`, return information on all child series. If `FALSE` (default), return only for the category `id`.
`n`	integer, maximum number of rows of series to return. Defaults to 50; maximum permitted by the API is 10,000.
`start`	integer, row to start from, defaults to 1.
`tidy`	logical, return a tidier result. See details.
`key`	API key: character if set explicitly; not needed if key is set globally. See `eia_set_key`.

Details

This function returns paginated results of the most recent update dates for data series. `n` and `start` help with stepping through chunks.

If you need to know the most recent update stamps for a large set of series, you should use this function, which makes an API call specifically to the EIA `updates` endpoint for specific EIA categories by category ID. If you are only interested in update times for a specific set of series IDs, you can use `eia_series_updates`. Note that while this function accepts a vector of IDs for `id`, it must make one API call per ID.

By default, additional processing is done to return a tibble data frame. Set `tidy = FALSE` to return only the initial list result of `jsonlite::fromJSON`. Set `tidy = NA` to return the original JSON as a character string.

Value

a tibble data frame (or a list, or character, depending on `tidy` value)

See Also

`eia_series_updates`

Examples

```
## Not run:
# use eia_set_key() to store stored API key
eia_updates(742, n = 5)

## End(Not run)
```

Index