Package: eia (via r-universe)

December 16, 2024

Title API Wrapper for U.S. Energy Information Administration ('EIA') Open Data

Version 0.4.2

Description Provides API access to data from the U.S. Energy Information Administration ('EIA') <https://www.eia.gov/>. Use of the EIA's API and this package requires a free API key obtainable at <https://www.eia.gov/opendata/register.php>. This package includes functions for searching the EIA data directory and returning time series and geoset time series datasets. Datasets returned by these functions are provided by default in a tidy format, or alternatively, in more raw formats. 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

URL https://docs.ropensci.org/eia/, https://github.com/ropensci/eia

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

Imports tibble, httr, jsonlite, memoise, lubridate

Suggests testthat (>= 3.0.0), knitr, covr, rmarkdown, dplyr, tidyr, ggplot2, spelling

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Config/pak/sysreqs libssl-dev

Repository https://ropensci.r-universe.dev

RemoteUrl https://github.com/ropensci/eia

RemoteRef master

RemoteSha 3c98e83381695f47fcbf2c7afc678d9372570660

Contents

`eia` *eia: EIA API wrapper*

Description

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

Author(s)

Maintainer: Matthew Hoff <matthew.g.hoff@gmail.com>

Authors:

- Matthew Leonawicz (ORCID)

See Also

Useful links:

- https://docs.ropensci.org/eia/
- https://github.com/ropensci/eia
- Report bugs at https://github.com/ropensci/eia/issues

`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.

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

Examples

```
eiadate_to_date(c("2018-03", "2018-04"))

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("2018-Q1", "2018-Q4"))
```

```
date_to_eiadate(x, "Q")
date_to_eiadate(x, "M")

(x <- eiadate_to_date("2019-01-02T16Z"))
date_to_eiadate(x, "H")
(x <- eiadate_to_date_seq("2019-01-02T16Z", "2019-01-02T19Z"))
date_to_eiadate(x, "H")
```

`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_dir()

eia_clear_metadata()

eia_clear_data()

eia_clear_facets()
```

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_dir(key))
system.time(eia_dir(key))
eia_clear_cache()
system.time(eia_dir(key))

## End(Not run)
```

`eia_data` *EIA data*

Description

Obtain data from the EIA.

Usage

```
eia_data(
  dir,
  data = NULL,
  facets = NULL,
  freq = NULL,
  start = NULL,
  end = NULL,
  sort = NULL,
  length = NULL,
  offset = NULL,
  tidy = TRUE,
  check_metadata = FALSE,
  cache = TRUE,
  key = eia_get_key()
)
```

Arguments

`dir`	character, directory path.
`data`	character or `NULL`, • note: if `NULL`, `eia_data()` will only return column headings; must input a character value as provided by `eia_metadata()` for data to be returned. • see details.
`facets`	character list or `NULL`, see details.
`freq`	character or `NULL`, see details.
`start, end`	character or `NULL`, see details.
`sort`	named list of two. • `cols`: list column names on which to sort. • `order`: `"asc"` or `"desc"` for ascending or descending, respectively.
`length`	numeric or `NULL`, number of rows to return.
`offset`	numeric or `NULL`, number of rows to skip before return.
`tidy`	logical or `NULL`, return a tidier result. See details.
`check_metadata`	logical, if `TRUE` checks input values against metadata endpoint.
`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, `data`, `facets`, and `freq` are set to `NULL`. To obtain valid input values for each of these arguments, use the specific ID labels as provided by `eia_metadata()`.

The use of `start` and `end` require some input to `freq`. By default (`check_metadata = FALSE`), the resulting data will match the temporal resolution provided to `freq`, however, `check_metadata = TRUE` applies further restrictions such that the format of values provided to `start`/`end` must match that of `freq`. Furthermore, regardless of the input format provided to `start`/`end`, the resulting data will always match the specification of `freq`. And lastly, regardless of chosen format, `end` must be strictly greater than `start` to return data.

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()`.

Value

data frame

Examples

```
## Not run:
eia_data(
  dir = "electricity/retail-sales",
  data = "price",
  facets = list(sectorid = c("COM", "RES"), stateid = "OH")
)

## End(Not run)
```

`eia_dir` *EIA directory*

Description

Obtain EIA directory listing.

Usage

```
eia_dir(dir = NULL, tidy = TRUE, cache = TRUE, key = eia_get_key())
```

Arguments

dir	character, directory path, if NULL then the API root directory.
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().

Value

data frame, list, or character; see details.

Examples

```
## Not run:
# use eia_set_key() to store API key
eia_dir()
eia_dir("electricity")
eia_dir("electricity/rto")

## End(Not run)
```

---

eia_facets *EIA facets*

Description

Obtain facets for a given set of EIA data.

Usage

```
eia_facets(dir, facet, tidy = TRUE, cache = TRUE, key = eia_get_key())
```

Arguments

dir	character, directory path.
facet	character
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()`.

Value

data frame, list, or character; see details.

Examples

```
## Not run:
eia_facets("electricity/retail-sales", facet = "sectorid")

## 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") # `NULL` with warning if not set where specified
```

`eia_metadata` *EIA metadata*

Description

Obtain EIA data metadata

Usage

```
eia_metadata(dir, tidy = TRUE, cache = TRUE, key = eia_get_key())
```

Arguments

`dir`	character, directory path.
`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()`.

Value

named list or character; see details.

Examples

```
## Not run:
eia_dir("electricity/retail-sales")
eia_metadata("electricity/retail-sales")

## End(Not run)
```

Index