Package 'octopusR'

June 9, 2023

Title Interact with the 'Octopus Energy' API

Version 1.0.1

Description A simple wrapper for the 'Octopus Energy' API <https://developer.octopus.energy/docs/api/>. It handles authentication, by storing a provided API key and meter details. Implemented endpoints include 'products' for viewing tariff details and 'consumption' for viewing meter consumption data.

License MIT + file LICENSE

URL https://github.com/Moohan/octopusR, https://moohan.github.io/octopusR/

BugReports https://github.com/Moohan/octopusR/issues

Imports askpass, cli, glue, httr2, rlang, tibble

Suggests covr, lubridate, spelling, testthat

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

RoxygenNote 7.2.3

NeedsCompilation no

Author James McMahon [aut, cre] (<https://orcid.org/0000-0002-5380-2029>)

Maintainer James McMahon <jamesmcmahon0@gmail.com>

Repository CRAN

Date/Publication 2023-06-09 15:10:07 UTC

R topics documented:

`get_consumption` *List consumption for a meter*

Description

Return a list of consumption values for half-hour periods for a given meter-point and meter.

Unit of measurement:

- Electricity meters: kWh
- SMETS1 Secure gas meters: kWh
- SMETS2 gas meters: m^3

Parsing dates:

To return dates properly parsed lubridate is required. Use the `tz` parameter to specify a time zone e.g. `tz = "UTC"`, the default (`tz = NULL`) will return the dates unparsed, as characters.

Usage

```
get_consumption(
  meter_type = c("electricity", "gas"),
  mpan_mprn = get_meter_details(meter_type)[["mpan_mprn"]],
  serial_number = get_meter_details(meter_type)[["serial_number"]],
  api_key = get_api_key(),
  period_from = NULL,
  period_to = NULL,
  tz = NULL,
  order_by = c("-period", "period"),
  group_by = c("hour", "day", "week", "month", "quarter")
)
```

Arguments

`meter_type`	Type of meter-point, electricity or gas
`mpan_mprn`	The electricity meter-point's MPAN or gas meter-point's MPRN.
`serial_number`	The meter's serial number.
`api_key`	Your API key. If you are an Octopus Energy customer, you can generate an API key on the developer dashboard.
`period_from`	Show consumption from the given datetime (inclusive). This parameter can be provided on its own.
`period_to`	Show consumption to the given datetime (exclusive). This parameter also requires providing the `period_from` parameter to create a range.
`tz`	a character string that specifies which time zone to parse the date with. The string must be a time zone that is recognized by the user's OS.

`order_by` Ordering of results returned. Default is that results are returned in reverse order from latest available figure. Valid values:

- `period`, to give results ordered forward.
- `-period`, (default), to give results ordered from most recent backwards.

`group_by` Aggregates consumption over a specified time period. A day is considered to start and end at midnight in the server's time zone. The default is that consumption is returned in half-hour periods. Accepted values are:

- `hour`
- `day`
- `week`
- `month`
- `quarter`

Value

a tibble of the requested consumption data.

`get_meter_gsp` *Get the GSP of a meter-point.*

Description

This endpoint can be used to get the GSP of a given meter-point.

Usage

```
get_meter_gsp(mpan = get_meter_details("electricity")[["mpan_mprn"]])
```

Arguments

`mpan` The electricity meter-point's MPAN

Value

a character of the meter-points GSP.

`get_products` *Return a list of energy products*

Description

By default, results will be public energy products but if authenticated organisations will also see products available to their organisation.

Usage

```
get_products(
  is_variable = NULL,
  is_green = NULL,
  is_tracker = NULL,
  is_prepay = NULL,
  is_business = FALSE,
  available_at = Sys.Date(),
  authenticate = FALSE,
  api_key = NULL
)
```

Arguments

`is_variable`	(boolean, optional) Show only variable products.
`is_green`	(boolean, optional) Show only green products.
`is_tracker`	(boolean, optional) Show only tracker products.
`is_prepay`	(boolean, optional) Show only pre-pay products.
`is_business`	(boolean, default: FALSE) Show only business products.
`available_at`	Show products available for new agreements on the given datetime. Defaults to current datetime, effectively showing products that are currently available.
`authenticate`	(boolean, default: FALSE) Use an `api_key` to authenticate. Only useful for organisations.
`api_key`	Your API key. If you are an Octopus Energy customer, you can generate an API key on the developer dashboard.

Value

a tibble

Examples

```
get_products(is_green = TRUE)
```

set_api_key *Set the Octopus API key*

Description

Set the Octopus API key to use. This will be stored as an environment variable. You should add `OCTOPUSR_API_KEY = <api_key>` to your `.Renviron` otherwise you will have to call this function every session.

Usage

```
set_api_key(api_key = NULL)
```

Arguments

`api_key`	Your API key. If you are an Octopus Energy customer, you can generate an API key on the developer dashboard.

Value

No return value, called for side effects.

set_meter_details *Set the details for your gas/electricity meter*

Description

Set the details for your gas/electricity meter. These will be stored as environment variables. You should add:

- `OCTOPUSR_MPAN = <electric MPAN>`
- `OCTOPUSR_MPRN = <gas MPRN>`
- `OCTOPUSR_ELEC_SERIAL_NUM = <electric serial number>`
- `OCTOPUSR_GAS_SERIAL_NUM = <gas serial number>` to your `.Renviron` otherwise you will have to call this function every session. You can find your meter details (MPAN/MPRN and serial number(s)) on the developer dashboard.

Usage

```
set_meter_details(
  meter_type = c("electricity", "gas"),
  mpan_mprn = NULL,
  serial_number = NULL
)
```

Arguments

`meter_type`	Type of meter-point, electricity or gas
`mpan_mprn`	The electricity meter-point's MPAN or gas meter-point's MPRN.
`serial_number`	The meter's serial number.

Value

No return value, called for side effects.

Index