Package 'reveneraR'

May 9, 2026

Title Connect to Your 'Revenera' (Formerly 'Revulytics') Data

Version 1.0.1

Description Facilitates making a connection to the 'Revenera' API and executing various queries. You can use it to get event data and metadata. The 'Revenera' documentation is available at <https://rui-api.redoc.ly/>. This package is not supported by 'Flexera' (owner of the software).

Suggests knitr, rmarkdown, ggplot2

Depends R (>= 4.3.0)

License CC0

URL https://github.com/chrisumphlett/reveneraR

BugReports https://github.com/chrisumphlett/reveneraR/issues

Encoding UTF-8

Imports dplyr (>= 1.1.4), magrittr (>= 2.0.3), jsonlite (>= 1.8.8), purrr (>= 1.0.2), httr (>= 1.4.7), tidyr (>= 1.3.1), rlang (>= 1.1.0)

RoxygenNote 7.3.2

NeedsCompilation no

Author Chris Umphlett [aut, cre], Avinash Panigrahi [aut]

Maintainer Chris Umphlett <christopher.umphlett@gmail.com>

Repository CRAN

Date/Publication 2025-02-25 18:50:02 UTC

Contents

`get_categories_and_events` *Get All Categories and Events for a List of Product Ids*

Description

Returns all of the unique categories and events (basic and advanced) for each product id.

Usage

```
get_categories_and_events(rev_product_ids)
```

Arguments

`rev_product_ids` A vector of Revenera product id's for which you want active user data.

Details

It is not recommended that your username be stored directly in your code. There are various methods and packages available that are more secure; this package does not require you to use any one in particular.

Value

Data frame with categories, events and event type by product id.

Examples

```
## Not run:
rev_user <- "my_username"
rev_pwd <- "super_secret"
logout(rev_user, rev_pwd)
Sys.sleep(30)
revenera_auth(rev_user, rev_pwd)
product_ids_list <- c("123", "456", "789")
session_id <- revenera_auth(rev_user, rev_pwd)
category_event <- get_categories_and_events(
  product_ids_list
)

## End(Not run)
```

`get_client_metadata` *Get Metadata on Client Ids for a List of Product Ids*

Description

Returns metadata (what Revenera calls "properties") for every Client Id installed during user-provided date range for all product Ids in a list.

Usage

```
get_client_metadata(
  rev_product_ids,
  product_properties_df,
  desired_properties,
  installed_start_date,
  installed_end_date,
  chatty = FALSE
)
```

Arguments

`rev_product_ids`	A vector of Revenera product id's for which you want active user data.
`product_properties_df`	Data frame with available properties for all product ids. Can obtain with the get_product_properties function.
`desired_properties`	The property names of the metadata you want to collect.
`installed_start_date`	Date object for the starting date of product installations.
`installed_end_date`	Date object for the ending date of product installations.
`chatty`	The function can be chatty, sending a message to the console for every iteration through a product Id. Many API calls may be required and the console may get very long and it may slow down the execution.

Details

It is not recommended that your username be stored directly in your code. There are various methods and packages available that are more secure; this package does not require you to use any one in particular.

This API call can only return 200 Client Ids at a time. It will take a long time to execute if you have many Client Ids, as the function will submit requests to the API repeatedly; this may even result in a timeout error from the server. In order to provide data for troubleshooting this function will write a message to the console after each call. It is recommended that you divert the console output to a

text file. You can do this in multiple ways, including with the sink function (see example for how to do this).

For the same reason you are encouraged to break your request into smaller chunks using the install dates and/or splitting up your product Ids.

Value

Data frame with selected properties for each Client Id.

Examples

```
## Not run:
rev_user <- "my_username"
rev_pwd <- "super_secret"
logout(rev_user, rev_pwd)
Sys.sleep(30)
revenera_auth(rev_user, rev_pwd)
product_ids_list <- c("123", "456", "789")
product_properties <- get_product_properties(product_ids_list)
sink("output_filename.txt") # write out chatty messages to a file
sink(stdout(), type = "message")
client_metadata <- get_client_metadata(
  product_ids_list, product_properties, c("Property1", "Property2"),
  start_date, end_date, chatty = TRUE
)
sink()

## End(Not run)
```

---

get_daily_client_properties

Get Daily Property Values for All Clients for a List of Product Ids

---

Description

Returns the list of daily client properties for all the client Ids installed during a user provided date range for all the Product Ids. In order to decrease how much is returned only the first date for a property value is returned. For example, if Property A had value 1 for 3 days, then value 2 for 2 days, then value 1 again on day 6, it will return the day 1 value of 1, day 4 value of 2, and day 6 value of 1.

Usage

```
get_daily_client_properties(
  rev_product_ids,
  product_properties_df,
  desired_properties,
```

```
  installed_start_date,
  installed_end_date,
  daily_start_date,
  daily_end_date,
  chatty = FALSE
)
```

Arguments

`rev_product_ids`	A vector of Revenera product id.
`product_properties_df`	Data frame with available properties for all product ids. Can obtain with the get_product_properties function.
`desired_properties`	The property names of the metadata you want to collect.
`installed_start_date`	Date object for the starting date of product installations.
`installed_end_date`	Date object for the ending date of product installations.
`daily_start_date`	Date object for the starting date of desired properties of the product.
`daily_end_date`	Date object for the ending date of desired properties of the product.
`chatty`	The function can be chatty, sending a message to the console for every iteration through a product Id. Many API calls may be required and the console may get very long and it may slow down the execution.

Details

It is not recommended that your username be stored directly in your code. There are various methods and packages available that are more secure; this package does not require you to use any one in particular.

This API call can only return 200 Client Ids at a time. It will take a long time to execute if you have many Client Ids, as the function will submit requests to the API repeatedly; this may even result in a timeout error from the server. In order to provide data for troubleshooting this function will write a message to the console after each call. It is recommended that you divert the console output to a text file. You can do this in multiple ways, including with the sink function (see example for how to do this).

For the same reason you are encouraged to break your request into smaller chunks using the install dates and/or splitting up your product Ids.

Value

Data frame with first date a property value appears until it changed for each Client Id.

Examples

```
## Not run:
rev_user <- "my_username"
rev_pwd <- "super_secret"
logout(rev_user, rev_pwd)
Sys.sleep(30)
revenera_auth(rev_user, rev_pwd)
product_ids_list <- c("123", "456", "789")
product_properties <- get_product_properties(product_ids_list)
sink("output_filename.txt") # write out chatty messages to a file
sink(stdout(), type = "message")
daily_client_properties <- get_daily_client_properties(product_ids_list,
  product_properties, c("Property1", "Property2"),
  installed_start_date = "01-01-2020", installed_end_date = "01-31-2020",
  daily_start_date = "01-01-2020", daily_end_date = "01-31-2020",
  chatty = TRUE
)
sink()

## End(Not run)
```

`get_product_properties` *Get All Properties for a List of Product Ids*

Description

Returns all of the unique properties (standard and custom) for each product id by property category.

Usage

```
get_product_properties(rev_product_ids)
```

Arguments

`rev_product_ids` A vector of Revenera product id's for which you want active user data.

Details

It is not recommended that your username be stored directly in your code. There are various methods and packages available that are more secure; this package does not require you to use any one in particular.

Value

Data frame with properties and property attributes by product id.

Examples

```
## Not run:
rev_user <- "my_username"
rev_pwd <- "super_secret"
logout(rev_user, rev_pwd)
Sys.sleep(30)
revenera_auth(rev_user, rev_pwd)
product_ids_list <- c("123", "456", "789")
product_properties <- get_product_properties(product_ids_list)

## End(Not run)
```

---

get_raw_data_files *Get Raw Data Files*

Description

Retrieves a list of raw data file exports that are available for a list of product IDs and the download URL for each file.

Usage

```
get_raw_data_files(rev_product_ids, days_back)
```

Arguments

rev_product_ids	A vector of Revenera product id's for which you want active user data.
days_back	How many days back to go to generate download URLs. Limiting this, if not all files are needed, will significantly reduce execution time.

Details

Raw data files are an add-on service available through Revenera. If these files are available they can be downloaded manually from the user portal, or downloaded via R. This function uses the API to first retrieve the list of files, and then get the download URL for each file.

It is not recommended that your username be stored directly in your code. There are various methods and packages available that are more secure; this package does not require you to use any one in particular.

Value

Data frame with available files and URLs.

Examples

```
## Not run:
rev_user <- "my_username"
rev_pwd <- "super_secret"
logout(rev_user, rev_pwd)
Sys.sleep(30)
revenera_auth(rev_user, rev_pwd)
product_ids_list <- c("123", "456", "789")
urls_df <- get_raw_data_files(product_ids_list, days_back = 3)
urls <- urls_df %>% pull(download_url)
file_names <-  urls_df %>%   pull(file_name)
file_list <- dplyr::pull(files_df, var = file_name)
dl_and_write <- function(u, f) {
  download.file(u, mode = "wb", destfile = f)
  upload_blob(cont, src = f, dest = paste0("/zip/", f))
  file.remove(f)
}
purrr::map2(urls, file_names, purrr::possibly(dl_and_write, "Download Error"))

## End(Not run)
```

---

get_users *Get Users by Product ID, Type, and Various Date Spans*

Description

For a given period of time (a day, week, or month) Revenera's API summarizes and returns the number of active users. With this function you can return daily, weekly, or monthly active users for multiple product ids.

Usage

```
get_users(
  rev_product_ids,
  user_type,
  rev_date_type,
  rev_start_date,
  rev_end_date,
  lost_days = 30,
  lost_reported = "dateLastSeen",
  group_by = "clientId",
  optional_json = ""
)
```

Arguments

`rev_product_ids`
: A vector of Revenera product id's for which you want active user data.

user_type	One of "active," "new", or "lost."
rev_date_type	Level of aggregation, Revenera will accept "day", "week", or "month".
rev_start_date	Date formatted YYYY-MM-DD. Revenera may give an error if you try to go back too far.
rev_end_date	Date formatted YYYY-MM-DD.
lost_days	Required for lost users, the number of consecutive days of inactivity before a client is considered lost.
lost_reported	Required for lost users, should the lost date be the first day of inactivity ("dateLastSeen") or date client is considered lost ("dateDeclaredLost").
group_by	Optional parameter, clientId by default, but can be changed to a different custom property that Revenera would use to group the data.
optional_json	Optional JSON text to add to the request body for things like global filters.

Details

You can specify a start and end date but Revenera does not store an indefinite period of historical data.

It is not recommended that your username be stored directly in your code. There are various methods and packages available that are more secure; this package does not require you to use any one in particular.

The optional_json parameter is available so that other optional arguments can be added to the api request. This will require the user to consult the API documentation. See README for an example.

Value

Data frame with active users for each product id and unique date within the range

Examples

```
## Not run:
rev_user <- "my_username"
rev_pwd <- "super_secret"
logout(rev_user, rev_pwd)
Sys.sleep(30)
revenera_auth(rev_user, rev_pwd)
product_ids_list <- c("123", "456", "789")
start_date <- lubridate::floor_date(Sys.Date(), unit = "months") - months(6)
end_date <- Sys.Date() - 1
global_filter <- paste0(
  ",\"globalFilters\":{\"licenseType\":",
  "{\"type\":\"string\",\"value\":\"purchased\"}}"
)
monthly_active_users <- get_users(product_ids_list,
  "active",
  "month",
  start_date,
  end_date,
  optional_json = global_filter
```

```
)

## End(Not run)
```

`logout` *Remove authorization cookies (logout).*

Description

Use this to remove the current authorization before re-authorizing.

Usage

```
logout(rev_username, rev_password)
```

Arguments

`rev_username`	Revenera username.
`rev_password`	Revenera password.

Details

It is not recommended that these values be stored directly in your code. There are various methods and packages available that are more secure; this package does not require you to use any one in particular.

Value

Nothing (successful logout), or an error message.

Examples

```
## Not run:
rev_user <- "my_username"
rev_pwd <- "super_secret"
revenera_auth(rev_user, rev_pwd)
logout(rev_user, rev_pwd)

## End(Not run)
```

`revenera_auth` *Login and Obtain Revenera API Session Id*

Description

An authorizaton cookie must first be established before querying for data. This is done using your Revenera username and password. If there is an active cookie this function will fail. You must 'logout()' first then

Usage

```
revenera_auth(rev_username, rev_password)
```

Arguments

`rev_username`	Revenera username.
`rev_password`	Revenera password.

Details

It is not recommended that these values be stored directly in your code. There are various methods and packages available that are more secure; this package does not require you to use any one in particular.

Value

Cookie authorization (which you won't see), or an error message.

Examples

```
## Not run:
rev_user <- "my_username"
rev_pwd <- "super_secret"
logout(rev_user, rev_pwd)
Sys.sleep(30)
revenera_auth(rev_user, rev_pwd)

## End(Not run)
```

Index