Package ‘tremendousr’

October 14, 2022

Title Easily Send Rewards and Incentives with 'Tremendous' from R

Version 1.0.0

Description A slightly-
opinionated R interface for the 'Tremendous' API (<https://www.tremendous.com/>).
In addition to supporting GET and POST requests, 'tremendousr' has, dare I say,
tremendously intuitive functions for sending digital rewards and incentives directly from R.

License MIT + file LICENSE

Encoding UTF-8

Language es

RoxygenNote 7.1.1.9000

Imports cli, crayon, crul, jsonlite

Suggests testthat (>= 3.0.0), clipr, vcr (>= 0.6.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Jonathan Trattner [aut, cre] (<https://orcid.org/0000-0002-1097-7603>),
Scott Chamberlain [ctb] (Utility functions `err_catcher()` and
`trem_ua()` from chimpr package.),
R Computing Lab at Wake Forest University [fnd]

Maintainer Jonathan Trattner <jdt@jdtrat.com>

Repository CRAN

Date/Publication 2021-09-30 12:20:02 UTC

R topics documented:

trem_client_new *Create a new Tremendous API Client*

Description

Create a new Tremendous API Client

Usage

```
trem_client_new(api_key = NULL, sandbox = TRUE)
```

Arguments

api_key	API key from tremendous.com. Can either pass in here as a character string or set for repeated use with `trem_set_api_key`.
sandbox	Logical: `TRUE` and any API requests are performed within the Tremendous sandbox environment, a free and fully-featured environment for application developing and testing. `FALSE` and the API requests are performed within the Tremendous production environment. **This will involve sending actual money, so be certain you wish to do this!**

Value

An object of class 'tremClient' that contains the API Key and Environment information for easily performing Tremendous API requests.

Examples

```
## Not run:

# Create a client for testing API calls within the Sandbox environment.
test_client <- trem_client_new(api_key = "TEST_YOUR-KEY-HERE",
                                  sandbox = TRUE)

# Create a client for performing API calls within a production environment.
# This client will be able to send actual money!
prod_client <- trem_client_new(api_key = "PROD_YOUR-KEY-HERE",
                               sandbox = FALSE)


## End(Not run)
```

`trem_delete` *Perform a DELETE request to Tremendous API*

Description

Tremendous only supports DELETE requests for one endpoint – deleting an invoice. Per their documentation, this request "removes an invoice. This has no further consequences but is a rather cosmetic operation." See the examples for a walk-through.

Usage

```
trem_delete(
  client,
  path,
  query = list(),
  body = NULL,
  disk = NULL,
  stream = NULL,
  encode = "json",
  parse = TRUE
)
```

Arguments

`client`	A Tremendous API Client object, created with `trem_client_new`.
`path`	The URL path, appended to the base URL, for GET requests such as listing available payment types, funding sources, account members, and more. See the Tremendous API Documentation for examples.
`query`	Query terms as a named list. See crul::HttpClient for more details.
`body`	Request body for Tremendous API, as an R List.
`disk`	A path to write to. `NULL` by default, so info is written to memory. See crul::HttpClient for more details.
`stream`	An R function to determine how to stream data. `NULL` by default, so info is streaned with memory. See crul::HttpClient for more details.
`encode`	"json" by default based on Tremendous API Request format. See crul::HttpClient for more options.
`parse`	Logical: Should the API Response results be parsed into a data frame?

Value

If `parse` = `TRUE` (default), a list containing the response from the API request. Otherwise, the R6 HttpResponse object containing API request data.

Examples

```
## Not run:

  # Create a new Tremendous API Client
  test_client <- trem_client_new(api_key = "TEST_YOUR-API-KEY-HERE",
                                  sandbox = TRUE)

  # Perform a POST request for an invoice.
  # `po_number` is Reference to the purchase order number within your organization
  # `amount` is in USD
  trem_post(test_client,
            path = "invoices",
            body = list(po_number = "unique-invoice-id",
                        amount = 50)
  )

  # Perform a GET request for listing all current (non-deleted) invoices.
  current_invoices <- trem_get(test_client, "invoices")

  # Get index for the correct ID
  unique_id_index <- which(current_invoices$invoices$po_number == "unique-invoice-id")

  # Get the invoice ID for 'unique-invoice-id' to delete
  my_invoice_id <- current_invoices$invoices[unique_id_index, "id"]

  # Perform a DELETE request for the specific invoice.
  trem_delete(test_client, paste0("invoices/", my_invoice_id))

  # Perform a GET request for listing all current (non-deleted) invoices.
  # The one with id po_number 'unique-invoice-id' should no longer be here.
  trem_get(test_client, "invoices")


## End(Not run)
```

trem_get *Perform a GET request to Tremendous API*

Description

This function provides lower-level access to perform GET requests via Tremendous API. Available endpoints can be found on the official Tremendous API documentation.

Usage

```
trem_get(
  client,
  path,
  query = list(),
  disk = NULL,
  stream = NULL,
  parse = TRUE
)
```

Arguments

`client` A Tremendous API Client object, created with `trem_client_new`.

`path` The URL path, appended to the base URL, for GET requests such as listing available payment types, funding sources, account members, and more. See the Tremendous API Documentation for examples.

`query` Query terms as a named list. See crul::HttpClient for more details.

`disk` A path to write to. `NULL` by default, so info is written to memory. See crul::HttpClient for more details.

`stream` An R function to determine how to stream data. `NULL` by default, so info is streaned with memory. See crul::HttpClient for more details.

`parse` Logical: Should the API Response results be parsed into a data frame?

Value

If `parse` = `TRUE` (default), a list containing the response from the API request. Otherwise, the R6 HttpResponse object containing API request data.

Examples

```
## Not run:

# Create a new Tremendous API Client
test_client <- trem_client_new(api_key = "TEST_YOUR-API-KEY-HERE",
sandbox = TRUE) # Sandbox environment so no actual money is sent

# Perform a GET request to list funding sources available in your Tremendous
# Account. Documentation:
# https://developers.tremendous.com/reference/core-funding-source-index
trem_get(trem_client, "funding_sources")

# Perform a GET request to list all invoices on your Tremendous Account.
# Documentation:
# https://developers.tremendous.com/reference/core-invoices-index
trem_get(trem_client, "invoices")

# Perform a GET request to list all orders (payment history) on your Tremendous
# Account. Documentation:
# https://developers.tremendous.com/reference/core-orders-index
trem_get(trem_client, "orders")
```

```
# Perform a GET request to list a specific order's information (payment history)
# from your Tremendous Account. Documentation:
# https://developers.tremendous.com/reference/core-orders-show
trem_get(trem_client, "orders/YOUR-ORDER-ID")


## End(Not run)
```

`trem_post` *Perform a POST request to Tremendous API*

Description

This function provides lower-level access to perform POST requests via Tremendous API. Available endpoints can be found on the official Tremendous API documentation.

For sending payments, I would recommend using `trem_send_reward` as it's more intuitive to use. However, this can be done using the `trem_post()` function (see examples).

Usage

```
trem_post(
  client,
  path,
  query = list(),
  body = NULL,
  disk = NULL,
  stream = NULL,
  encode = "json",
  parse = TRUE
)
```

Arguments

`client`	A Tremendous API Client object, created with `trem_client_new`.
`path`	The URL path, appended to the base URL, for GET requests such as listing available payment types, funding sources, account members, and more. See the Tremendous API Documentation for examples.
`query`	Query terms as a named list. See crul::HttpClient for more details.
`body`	Request body for Tremendous API, as an R List.
`disk`	A path to write to. `NULL` by default, so info is written to memory. See crul::HttpClient for more details.
`stream`	An R function to determine how to stream data. `NULL` by default, so info is streaned with memory. See crul::HttpClient for more details.
`encode`	"json" by default based on Tremendous API Request format. See crul::HttpClient for more options.
`parse`	Logical: Should the API Response results be parsed into a data frame?

Value

If parse = TRUE (default), a list containing the response from the API request. Otherwise, the R6 HttpResponse object containing API request data.

Examples

```
## Not run:

  # Create a new Tremendous API Client
  test_client <- trem_client_new(api_key = "TEST_YOUR-API-KEY-HERE",
                                 sandbox = TRUE)

  # Perform a POST request to invite new members to your Tremendous Account.
  # Documentation: https://developers.tremendous.com/reference/post_members
    trem_post(trem_client,
              path = "members",
              body = list(email = "example@website.com",
                          name = "Example Person",
                          role = "MEMBER"))

  # Perform a POST send payments --
  I find it ~tremendously~ easier to use the `trem_send_reward()` function.
  # Documentation: https://developers.tremendous.com/reference/core-orders-create
    trem_post(trem_client,
              path = "orders",
              body = list(
                external_id = "manual-payment-post", # This is a payment description id
                payment = list(
                  funding_source_id = "your-funding-id-from-tremendous"
                ),
                rewards = list(
                  value = list(
                    denomination = 10,
                    currency_code = "USD"
                  ),
                  delivery = list(
                    method = "EMAIL" # "EMAIL", "LINK", or "PHONE",
                  ),
                  recipient = list(
                    name = "first last",
                    email = "email@website.com"
                  ),
                  # IDs for Applebee's Gift Card and Amazon Gift Card
                  products = c("2JFKPXBWDC1K", "VW9JLMPRL9N7")
                )
              ))


## End(Not run)
```

trem_send_reward *Send a Reward via Tremendous API*

Description

The most likely reason to use the tremendousr package is to send rewards This function, `trem_send_reward()`, provides an easy interface to do so. See the examples for more details.

Usage

```
trem_send_reward(
  client,
  name,
  email = NULL,
  phone = NULL,
  reward_amount,
  currency_code = "USD",
  delivery_method = "EMAIL",
  payment_description_id,
  funding_source_id,
  reward_types,
  parse = TRUE
)
```

Arguments

`client` A Tremendous API Client object, created with `trem_client_new`.

`name` Name of the recipient.

`email` Email address of the recipient.

`phone` Phone number of the recipient (US phone numbers only).

`reward_amount` Amount of the reward (numeric).

`currency_code` Currency of the reward (default to "USD").

`delivery_method` Default to "EMAIL", for sending the reward to the recipient via email. Alternatively, reward can be delivered via a link ("LINK") or text message ("PHONE").

`payment_description_id` Unique ID for specific order. This will appear as `external_id` on Tremendous Dashboard.

`funding_source_id` ID of the funding source linked to your account, to draw funds from for this order. One of the IDs from `trem_get("funding_sources")`.

`reward_types` A character vector of product ids – reward options – for the recipient to choose from. Available options can be found here.

`parse` Logical: Should the API Response results be parsed into a data frame?

Value

If parse = TRUE (default), a list containing the response from payment API request. Otherwise, the R6 HttpResponse object containing API request data.

Examples

```
## Not run:

  # Create a Tremendous Client
  test_client <- trem_client_new(api_key = "TEST_YOUR-KEY-HERE",
                          sandbox = TRUE) # Sandbox environment so no actual money is sent

  # To send a payment, you can simply pass in the client
  # and specify the necessary fields.
  payment1 <- trem_send_reward(client = test_client,
                           name = "first last",
                           email = "email@website.com",
                           reward_amount = 10,
                           currency_code = "USD",
                           delivery_method = "EMAIL",
                           payment_description_id = "payment-from-tremendousr-examples",
                           funding_source_id = "your-funding-id-from-tremendous",
                         reward_types = "Q24BD9EZ332JT", # ID for virtual visa gift card
                           parse = TRUE # Return a parsed API response
  )


## End(Not run)
```

---

trem_set_api_key *Set Tremendous API Key in .Renviron*

Description

Set Tremendous API Key in .Renviron

Usage

```
trem_set_api_key(api_key)
```

Arguments

api_key	API key from tremendous.com.

Value

NA; Used to set the Tremendous API key for persistent use in '.Renviron'.

Examples

```
if (interactive()) {
  # For Sandbox Environment, your API key should look like:
  trem_set_api_key("TEST_YOUR-API-KEY")

  # For Production Environment, your API key should look like:
  trem_set_api_key("PROD_YOUR-API-KEY")
}
```

---

trem_ua *Get useragent info for tremendousr API Package*

Description

Exported for use with `trem_client_new` to provide useragent info as a curl option for the API Request. This function was Adapted from chimpr.

Usage

```
trem_ua()
```

Value

useragent info for tremendousr API Package

Examples

```
# Get useragent info for the tremendous API package
trem_ua()
```

Index