Package 'LendingClub'

June 15, 2017

Type Package

Date 2017-06-16

Title A Lending Club API Wrapper

Version 1.0.3

URL https://github.com/kuhnrl30/LendingClub

BugReports https://github.com/kuhnrl30/LendingClub/issues

Description Functions to access Lending Club's API and assist the investor manage their account. Lending Club is a peer-to-peer lending service where loans are broken up into $25 notes that investors buy with the expectation of earning a return on the interest. You can learn more about the API here: <http://www.lendingclub.com/developers/lc-api.action>.

License MIT + file LICENSE

LazyData TRUE

BuildVignettes TRUE

Suggests knitr

VignetteBuilder knitr

RoxygenNote 6.0.1

Imports dplyr, plyr, httr, jsonlite, utils

NeedsCompilation no

Author Ryan Kuhn [aut, cre]

Maintainer Ryan Kuhn <kuhnrl30@gmail.com>

Repository CRAN

Date/Publication 2017-06-15 13:39:50 UTC

R topics documented:

`AccountSummary` *Return the investor's Account Summary*

Description

Access the user's account and return the summary data. The data points include available cash, total account balance, and the number of notes in the portfolio.

Usage

```
AccountSummary(LC_CRED = NULL)
```

Arguments

`LC_CRED`	Credential object having class LendingClub_credential. This object can be created using the `MakeCredential()` function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".

Value

Object of class LendingClub_api. The object has two components. The first is `$content` with a data.frame of attributes of the invenstor's account such as available cash and value of the portfolios. The attributes are arranged by rows in the first column and the values are in the second column. The second component is the `$response` including a timestamp of when the data was retrieved.

Examples

```
## Not run:
LC_CRED<- MakeCredential(InvestorID, APIKey)
AccountSummary()
## End(Not run)
```

AddFunds *Add funds to the investor's account*

Description

Sets up a recurring or one-time funds transfer. The investor must have already setup the bank account from which the funds are transferred. Transfers can one-time or recurring.

Usage

```
AddFunds(amount, freq = "LOAD_NOW", start = NULL, end = NULL, quiet = T,
  LC_CRED = NULL)
```

Arguments

amount	Numeric. Amount to be transferred
freq	String. Frequency of transfers. Optional. The default value is LOAD_NOW so if a value is not supplied the transfer will be processed immediately. If a value is supplied, it must me one of: • LOAD_NOW (default) • LOAD_ONCE • LOAD_WEEKLY • LOAD_BIWEEKLY • LOAD_ON_DAY_1_AND_15 • LOAD_MONTHLY
start	String. Defaults to NULL and is not required for immediate transfers. For future transfers, the date must be a string formatted as YYYY/MM/DD.
end	String. Defaults to NULL and is not required for immediate transfers. For future transfers, the date must be a string formatted as YYYY/MM/DD.
quiet	Should the response be printed to the console
LC_CRED	Credential object having class LendingClub_credential. This object can be created using the `MakeCredential()` function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".

AvailableCash *Return investor's available cash*

Description

Uses Lending Club's API to access the user's account and return the available cash. Committed cash is excluded from the total.

Usage

```
AvailableCash(LC_CRED = NULL)
```

Arguments

LC_CRED	Credential object having class LendingClub_credential. This object can be created using the MakeCredential() function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".

CancelTransfer *Cancel a scheduled transfer*

Description

Cancel a scheduled transfer. You can get a listing of scheduled transfers by using the PendingTransfers() funciton.

Usage

```
CancelTransfer(transferId, LC_CRED = NULL)
```

Arguments

transferId	Numeric. Unique ID for the scheduled transfer
LC_CRED	Credential object having class LendingClub_credential. This object can be created using the MakeCredential() function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".

CreatePortfolio *Create a new portfolio*

Description

Create a new portfolio to assign loans to. Notes can be assigned to a portfolio using the API when they are purhcased from the primary market.

Usage

```
CreatePortfolio(port_name, port_desc, LC_CRED = NULL, quiet = T)
```

Arguments

port_name	String. Name of the new portfolio
port_desc	String. Portfolio description
LC_CRED	Credential object having class LendingClub_credential. This object can be created using the MakeCredential() function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".
quiet	Should the response be printed to the console

DetailedNotesOwned *Listing of investor's loans*

Description

Access the user's account and return a dataframe of the investor's notes owned.

Usage

```
DetailedNotesOwned(LC_CRED = NULL)

NotesOwned(LC_CRED = NULL)
```

Arguments

LC_CRED	Credential object having class LendingClub_credential. This object can be created using the MakeCredential() function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".

Details

The DetailedNotesOwned) function returns nearly the same variables as NotesOwned(). As the name suggests, the function returns additional information about the notes owned including datapoints on the purpose, portfolio, and the payment history.

FolioBuy *Transact with the secondary market*

Description

Buy or sell notes on the FOLIOfn secondary market

Usage

```
FolioBuy(loanId, orderId, noteId, Price = NULL, LC_CRED = NULL, quiet = T)

FolioSell(loanId, orderId, noteId, expireDate = NULL, Price = NULL,
  LC_CRED = NULL, quiet = T)
```

Arguments

loanId	The loan ID
orderId	The loan's order ID
noteId	The loan's note ID
Price	Price you wish to sell your loan for. Remember to consider
LC_CRED	Credential object having class LendingClub_credential. This object can be created using the MakeCredential() function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".
quiet	Should the response be printed to the console the accrued interest in your offering.
expireDate	The date the sell offer will expire. Can be as short as the current date or up to 7 days from posting the offer. Character string.

FolioListing *Loans listed on Folio marketplace*

Description

Returns a dataframe of loans listed on the secondary market.

Usage

```
FolioListing()
```

LendingClub *Lending Club*

Description

Providing the tools to work with the Lending Club API. Both the primary and secondary markets can be accessed with this package.

Details

Lending Club is a peer-to-peer lending service where loans are broken up into $25 notes that investors buy with expectation of earing a return on the interest. This package is built to assist the investor manage their account and maximize their return.

To learn more about the LendingClub package, start with the vignette: `vignette("LendingClub")`

Author(s)

Ryan Kuhn, CPA

ListedLoans *Listing of all loans on the primary market*

Description

Uses Lending Club's API to return all current listed loans

Usage

```
ListedLoans(showAll = TRUE, LC_CRED = NULL)
```

Arguments

`showAll`	logical, indicate whether all loans should be retured. Use false to limit the results to those loans listed in the most recent listing period.
`LC_CRED`	Credential object having class LendingClub_credential. This object can be created using the `MakeCredential()` function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".

Value

LendingClub_api object. There are two components, the first is the content of the request, the second is the response from the API. The content can be further subdivided into a 2 element list. The first element is the date of the request and second is a dataframe of the listed loans.

MakeCredential *Create the Lending Club credential*

Description

Creates the credential needed to access the other functions in this package.

Usage

```
MakeCredential(investorID, APIkey)
```

Arguments

investorID	User's ID from the Account Summary page
APIkey	User's API key from the account settings page
...	See the details section for more information.

Details

This is the first step to accessing the API. Pass the function your investor ID and API key. The investor ID can be found on your account summary page and the API Key can be found on your account settings page. The function will create an object of class LendingClub_credential. If the credential is stored to the variable LC_CRED, then the other functions in the package will be able to read the credential from the global environment. Else, you may pass the credential to the other functions as an argument.

Examples

```
LC_CRED<- MakeCredential("investorID", "APIkey")
```

PendingTransfers *Return investor's pending transfers*

Description

Uses Lending Club's API to access the user's account and return the summary

Usage

```
PendingTransfers(LC_CRED = NULL)
```

Arguments

LC_CRED	Credential object having class LendingClub_credential. This object can be created using the MakeCredential() function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".

PortfoliosOwned *Listing of investor's portfolios*

Description

Provides a dataframe of all portfolios owned by the investor.

Usage

```
PortfoliosOwned(LC_CRED = NULL)
```

Arguments

LC_CRED	Credential object having class LendingClub_credential. This object can be created using the MakeCredential() function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".

SubmitOrder *Buy a note on the primary market*

Description

Create an order to purchase a note. If the account has sufficient funding, the note will be purchased once the loan has passed through the funding stage.

Usage

```
SubmitOrder(loanId, amount = 25, portfolioId = NULL, LC_CRED = NULL,
  quiet = T)
```

Arguments

loanId	Unique ID for loans. These can be found using the ListedLoans() function.
amount	Amount to be purchased
portfolioId	Unique ID for the portfolio. It can be found using the PortfoliosOwned() function.
LC_CRED	Credential object having class LendingClub_credential. This object can be created using the MakeCredential() function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".
quiet	Should the response be printed to the console

Examples

```
## Not run:
SubmitOrder(12345678, 25)
SubmitOrder(12345678, 25, "myPortfolio")

#For multiple loans in a single order
SubmitOrder(loanId=c(123, 456, 789), amount= c(rep(25,3)))

## End(Not run)
```

`WithdrawFunds` *Withdraw funds*

Description

Withdraw funds from the investor's account

Usage

```
WithdrawFunds(amount, LC_CRED = NULL)
```

Arguments

`amount`	Amount to be transferred
`LC_CRED`	Credential object having class LendingClub_credential. This object can be created using the `MakeCredential()` function. You can pass the object directly as an argument to the function or alternatively, it can be read from the global environment if it was stored as "LC_CRED".

Index