

Package ‘riingo’

May 11, 2019

Type Package

Title An R Interface to the 'Tiingo' Stock Price API

Version 0.2.0

Maintainer Davis Vaughan <davis@rstudio.com>

Description Functionality to download stock prices, cryptocurrency data, and more from the 'Tiingo' API <<https://api.tiingo.com/>>.

URL <https://github.com/business-science/riingo>

BugReports <https://github.com/business-science/riingo/issues>

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 6.1.1

LazyData true

Depends R (>= 2.10)

Imports crayon (>= 1.3.4), dplyr (>= 0.7.4), glue (>= 1.3.1), httr (>= 1.3.1), jsonlite (>= 1.5), tibble (>= 1.4.2), rlang (>= 0.2.0), purrr (>= 0.2.4)

Suggests testthat, covr

NeedsCompilation no

Author Davis Vaughan [aut, cre],
Matt Dancho [aut]

Repository CRAN

Date/Publication 2019-05-11 06:40:03 UTC

R topics documented:

convert_to_local_time	2
is_supported_ticker	2
riingo_browse_usage	3
riingo_crypto_latest	4

riingo_crypto_meta	5
riingo_crypto_prices	5
riingo_crypto_quote	7
riingo_iex_latest	8
riingo_iex_prices	9
riingo_iex_quote	10
riingo_latest	11
riingo_meta	11
riingo_prices	12
riingo_set_token	13

Index	15
--------------	-----------

`convert_to_local_time` *Convert the POSIXct columns of a data frame to the local time zone*

Description

Tiingo returns data with a UTC time zone. Often you will want to view this in your own time zone. This function converts each POSIXct column of the returned tibble to the local (or specified) time zone.

Usage

```
convert_to_local_time(.data, tz = "")
```

Arguments

<code>.data</code>	A tibble with POSIXct columns
<code>tz</code>	The time zone to convert to. The default is the local time zone.

`is_supported_ticker` *Ticker information*

Description

`is_supported_ticker()` can tell you if a given ticker is supported on Tiingo. `supported_tickers()` returns a tibble listing every available ticker.

Usage

```
is_supported_ticker(ticker, type = "tiingo")
```

```
supported_tickers(type = "tiingo")
```

Arguments

ticker	The single ticker to check for on Tiingo.
type	One of: "tiingo", "iex", or "crypto".

Examples

```
## Not run:

# VOO is supported on both Tiingo and IEX
is_supported_ticker("VOO")
is_supported_ticker("VOO", type = "iex")

# PRHSX is a mutual fund that is supported by Tiingo but not IEX
is_supported_ticker("PRHSX")
is_supported_ticker("PRHSX", type = "iex")

# BTCUSD is available
is_supported_ticker("btcusd", type = "crypto")

## End(Not run)
```

riingo_browse_usage *Browse various pages of the Tiingo site*

Description

Note that you **must** be signed into the site on the opened browser for most of these functions to work properly, otherwise you will be redirected to the sign in page.

Usage

```
riingo_browse_usage()

riingo_browse_token()

riingo_browse_documentation()

riingo_browse_signup()
```

riingo_crypto_latest *The latest day's worth of intraday data for a given cryptocurrency*

Description

This returns only the most recent day of intraday data for the supplied ticker(s).

Usage

```
riingo_crypto_latest(ticker, resample_frequency = "1min",
  base_currency = NULL, exchanges = NULL, convert_currency = NULL,
  raw = FALSE)
```

Arguments

ticker	One or more cryptocurrency tickers. Specified as "btcusd" for bitcoin quoted in USD. A character vector.
resample_frequency	For Tiingo data, a character specified as one of: "daily", "monthly", "quarterly" or "yearly". For IEX data, a character specified at the "min" or "hour" frequencies in the form: "1min", "5min", or "2hour". For Crypto data, a character specified at the "min", "hour" or "day" frequencies similar to IEX.
base_currency	<i>Instead of ticker</i> you may pass a base currency. This selects all currencies with that base currency. For example if 'base_currency="btc"' tickers <i>btcusd, btcjpy, btceur</i> , etc.. will all be returned.
exchanges	If you would like to limit the query to a subset of exchanges, pass a comma-separated list of exchanges to select. Example) "POLONIEX, GDAX"
convert_currency	This parameter will convert the return data into another fx rate. For example if querying BTCUSD and convert_currency is 'cure', the bitcoin prices will be converted to CureCoin prices. Setting this to a value will add fxOpen, fxHigh, fxLow, fxClose, fxVolumeNotional, and fxRate accordingly. fxRate is the rate used to perform the currency calculation. If exchanges is specified, the conversion rate will be calculated using the exchanges passed.
raw	If TRUE, the raw underlying data from multiple exchanges will be returned, rather than the clean prices. This is the data that calculates the aggregated prices and quotes.

Examples

```
## Not run:

# The latest available day of intraday data for QQQ
riingo_crypto_latest("btcusd")
```

```
## End(Not run)
```

```
riingo_crypto_meta    Get meta data about a cryptocurrency on Tiingo
```

Description

Relevant returned meta data include: ticker, name, description, quote currency, and base currency.

Usage

```
riingo_crypto_meta(ticker)
```

Arguments

ticker One or more cryptocurrency tickers. Specified as "btcusd" for bitcoin quoted in USD. A character vector.

Examples

```
## Not run:  
  
# Bitcoin meta  
riingo_crypto_meta("btcusd")  
  
# A trick to return ALL crypto meta data  
# For some reason Descriptions are not returned here  
riingo_crypto_meta("")  
  
## End(Not run)
```

```
riingo_crypto_prices  Get cryptocurrency prices aggregated through Tiingo
```

Description

Get cryptocurrency prices aggregated through Tiingo

Usage

```
riingo_crypto_prices(ticker, start_date = NULL, end_date = NULL,
  resample_frequency = "1day", base_currency = NULL,
  exchanges = NULL, convert_currency = NULL, raw = FALSE)
```

Arguments

ticker	One or more cryptocurrency tickers. Specified as "btcusd" for bitcoin quoted in USD. A character vector.
start_date	The first date to download data for. A character in the form YYYY-MM-DD, or a Date variable. The default is to download 1 year's worth of data.
end_date	The last date to download data for. A character in the form YYYY-MM-DD, or a Date variable.
resample_frequency	For Tiingo data, a character specified as one of: "daily", "monthly", "quarterly" or "yearly". For IEX data, a character specified at the "min" or "hour" frequencies in the form: "1min", "5min", or "2hour". For Crypto data, a character specified at the "min", "hour" or "day" frequencies similar to IEX.
base_currency	Instead of ticker you may pass a base currency. This selects all currencies with that base currency. For example if 'base_currency="btc"' tickers <i>btcusd</i> , <i>btcjpy</i> , <i>btceur</i> , etc.. will all be returned.
exchanges	If you would like to limit the query to a subset of exchanges, pass a comma-separated list of exchanges to select. Example) "POLONIEX, GDAX"
convert_currency	This parameter will convert the return data into another fx rate. For example if querying BTCUSD and convert_currency is 'cure', the bitcoin prices will be converted to CureCoin prices. Setting this to a value will add fxOpen, fxHigh, fxLow, fxClose, fxVolumeNotional, and fxRate accordingly. fxRate is the rate used to perform the currency calculation. If exchanges is specified, the conversion rate will be calculated using the exchanges passed.
raw	If TRUE, the raw underlying data from multiple exchanges will be returned, rather than the clean prices. This is the data that calculates the aggregated prices and quotes.

Examples

```
## Not run:

# Bitcoin prices
riingo_crypto_prices("btcusd")

# Bitcoin in USD and EUR
riingo_crypto_prices(c("btcusd", "btceur"), start_date = "2018-01-01", resample_frequency = "5min")

# Bitcoin raw data
riingo_crypto_prices("btcusd", raw = TRUE)
```

```
# Only use the POLONIEX exchange
riingo_crypto_prices("btcusd", raw = TRUE, exchanges = "POLONIEX")

# All btc___ cryptocurrency pairs
riingo_crypto_prices(base_currency = "btc")

## End(Not run)
```

riingo_crypto_quote *Quote and Top of Book data for a given cryptocurrency*

Description

Tiingo provides TOP (top of book) bid and ask quotes for cryptocurrencies. Note that this cannot be historically queried.

Usage

```
riingo_crypto_quote(ticker, exchanges = NULL, convert_currency = NULL,
  raw = FALSE)
```

Arguments

ticker	One or more cryptocurrency tickers. Specified as "btcusd" for bitcoin quoted in USD. A character vector.
exchanges	If you would like to limit the query to a subset of exchanges, pass a comma-separated list of exchanges to select. Example) "POLONIEX, GDAX"
convert_currency	This parameter will convert the return data into another fx rate. For example if querying BTCUSD and convert_currency is 'cure', the bitcoin prices will be converted to CureCoin prices. Setting this to a value will add fxOpen, fxHigh, fxLow, fxClose, fxVolumeNotional, and fxRate accordingly. fxRate is the rate used to perform the currency calculation. If exchanges is specified, the conversion rate will be calculated using the exchanges passed.
raw	If TRUE, the raw underlying data from multiple exchanges will be returned, rather than the clean prices. This is the data that calculates the aggregated prices and quotes.

Details

At the end of the day, the, askPrice, bidSize, bidPrice, askSize, and lastSize fields may be NA. This is normal.

Examples

```
## Not run:

riingo_crypto_quote("btcusd")

# The raw data can provide more insight into each individual exchange
riingo_crypto_quote("btcusd", raw = TRUE)

## End(Not run)
```

riingo_iex_latest	<i>The latest day's worth of intraday data for a given ticker</i>
-------------------	---

Description

This returns only the most recent day of intraday data for the supplied ticker(s).

Usage

```
riingo_iex_latest(ticker, resample_frequency = "1min")
```

Arguments

ticker	One or more tickers to download data for from Tiingo. Can be a stock, mutual fund, or ETF. A character vector.
resample_frequency	For Tiingo data, a character specified as one of: "daily", "monthly", "quarterly" or "yearly". For IEX data, a character specified at the "min" or "hour" frequencies in the form: "1min", "5min", or "2hour". For Crypto data, a character specified at the "min", "hour" or "day" frequencies similar to IEX.

Examples

```
## Not run:

# The latest available day of intraday data for QQQ
riingo_iex_latest("QQQ")

riingo_iex_latest("QQQ", "1hour")

## End(Not run)
```

riingo_iex_prices *Get stock or ETF prices from IEX through Tiingo*

Description

The Tiingo API provides a way to access data from IEX, The Investors Exchange. This data is supplied at a much lower (intraday!) frequency than the data from Tiingo's native API.

Usage

```
riingo_iex_prices(ticker, start_date = NULL, end_date = NULL,
  resample_frequency = "5min")
```

Arguments

ticker	One or more tickers to download data for from Tiingo. Can be a stock, mutual fund, or ETF. A character vector.
start_date	The first date to download data for. A character in the form YYYY-MM-DD, or a Date variable. The default is to download 1 year's worth of data.
end_date	The last date to download data for. A character in the form YYYY-MM-DD, or a Date variable.
resample_frequency	For Tiingo data, a character specified as one of: "daily", "monthly", "quarterly" or "yearly". For IEX data, a character specified at the "min" or "hour" frequencies in the form: "1min", "5min", or "2hour". For Crypto data, a character specified at the "min", "hour" or "day" frequencies similar to IEX.

Details

This feed returns the most recent 2000 ticks of data at the specified frequency. For example, "5min" would return the 2000 most recent data points spaced 5 minutes apart. You can subset the returned range with `start_date` and `end_date`, but **you cannot request data older than today's date minus 2000 data points.**

Because the default attempts to pull 1 year's worth of data, at a 5 minute frequency, all available data will be pulled so there is no need to use `start_date` and `end_date`. Only use them if you set the frequency to hourly.

Examples

```
## Not run:

# Pulling all available minute level data for Apple
riingo_iex_prices("AAPL", resample_frequency = "1min")

# This would result in an error, as you are pulling outside the available range
# riingo_iex_prices("AAPL", "1990-01-01", "2000-01-01", resample_frequency = "5min")
```

```
## End(Not run)
```

riingo_iex_quote	<i>Quote and Top of Book data for a given ticker</i>
------------------	--

Description

Tiingo is plugged into the IEX feed, and they provide last sale data along with TOP (top of book) bid and ask quotes. Note that this cannot be historically queried.

Usage

```
riingo_iex_quote(ticker)
```

Arguments

ticker	One or more tickers to download data for from Tiingo. Can be a stock, mutual fund, or ETF. A character vector.
--------	--

Details

At the end of the day, the mid, askPrice, bidSize, bidPrice, askSize, and lastSize fields will be NA. This is normal.

Examples

```
## Not run:  
  
riingo_iex_quote("QQQ")  
  
## End(Not run)
```

riingo_latest	<i>The latest day's worth of data for a given ticker</i>
---------------	--

Description

This returns only the most recent day of daily data for the supplied ticker(s).

Usage

```
riingo_latest(ticker)
```

Arguments

ticker	One or more tickers to download data for from Tiingo. Can be a stock, mutual fund, or ETF. A character vector.
--------	--

Examples

```
## Not run:  
  
# The latest available day of daily data for QQQ  
riingo_latest("QQQ")  
  
## End(Not run)
```

riingo_meta	<i>Get meta data about a ticker available on Tiingo</i>
-------------	---

Description

Retrieve start and end dates for available ticker data, along with the name, exchange, and description of the ticker.

Usage

```
riingo_meta(ticker)
```

Arguments

ticker	One or more tickers to download data for from Tiingo. Can be a stock, mutual fund, or ETF. A character vector.
--------	--

Examples

```
## Not run:

riingo_meta("AAPL")

riingo_meta("QQQ")

## End(Not run)
```

riingo_prices	<i>Get stock or ETF prices from the Tiingo API</i>
---------------	--

Description

The Tiingo API provides a large feed of historical data at the daily (and monthly, quarterly, or yearly) level.

Usage

```
riingo_prices(ticker, start_date = NULL, end_date = NULL,
              resample_frequency = "daily")
```

Arguments

ticker	One or more tickers to download data for from Tiingo. Can be a stock, mutual fund, or ETF. A character vector.
start_date	The first date to download data for. A character in the form YYYY-MM-DD, or a Date variable. The default is to download 1 year's worth of data.
end_date	The last date to download data for. A character in the form YYYY-MM-DD, or a Date variable.
resample_frequency	For Tiingo data, a character specified as one of: "daily", "monthly", "quarterly" or "yearly". For IEX data, a character specified at the "min" or "hour" frequencies in the form: "1min", "5min", or "2hour". For Crypto data, a character specified at the "min", "hour" or "day" frequencies similar to IEX.

Details

Multiple downloads are done *sequentially*, meaning that downloading 5 tickers costs 5 requests against your usage limits. Sadly Tiingo does not support batch downloads at the moment.

Tiingo supplied documentation regarding the resample frequency:

- daily: Values returned as daily periods, with a holiday calendar
- weekly: Values returned as weekly data, with days ending on Friday

- **monthly:** Values returned as monthly data, with days ending on the last standard business day (Mon-Fri) of each month
- **annually:** Values returned as annual data, with days ending on the last standard business day (Mon-Fri) of each year
- Note, that if you choose a value in-between the resample period for weekly, monthly, and daily, the start date rolls back to consider the entire period. For example, if you choose to resample weekly, but your "start_date" parameter is set to Wednesday of that week, the start_date will be adjusted to Monday, so the entire week is captured. Another example is if you send a start_date mid-month, we roll back the start_date to the beginning of the month.
- Similarly, if you provide an end_date, and it's midway through the period, we roll-forward the date to capture the whole period. In the above example, if the end date is set to a wednesday with a weekly resample, the end date is rolled forward to the Friday of that week.

Examples

```
## Not run:  
  
# Downloading 1 year's worth of prices for AAPL  
riingo_prices("AAPL")  
  
# Downloading a range of data, using 2 tickers  
riingo_prices(c("AAPL", "MSFT"), "1999-01-01", "2005-01-01")  
  
# Monthly data  
riingo_prices(c("AAPL", "MSFT"), "1999-01-01", "2005-01-01", "monthly")  
  
## End(Not run)
```

riingo_set_token *Set and get your Tiingo API token*

Description

There are two methods for setting your token, an environment variable in your .Renviron file, or setting an option. If both are set, the environment variable will always be used. See details for how to get started.

Usage

```
riingo_set_token(token)  
  
riingo_get_token()
```

Arguments

token Tiingo API token. A character.

Details

To use the Tiingo API, you must create an account and set an API token. It is completely free to get started and use their free source of data.

To sign up, use [riingo_browse_signup\(\)](#) and click Sign-up.

To find your API token, use [riingo_browse_token\(\)](#). Note that you must be signed in on the opened browser.

With your API token in hand, you can do one of two things:

- Set the API token with [riingo_set_token\(\)](#). This is only valid for the current R session and must be done each time you open R.
- Set the API token as the RIINGO_TOKEN environment variable in an `.Renviro`n file. This is what I recommend. The easiest way to access this file is with the `usethis` package. Open it with `usethis::edit_r_enviro`n() and then add a line with `RIINGO_TOKEN = token_here`. Do not put the token in quotes, and make sure to restart R once you have set it. After that, you shouldn't have to worry about it again.

Index

`convert_to_local_time`, 2

`is_supported_ticker`, 2
`is_supported_ticker()`, 2

`riingo_browse_documentation`
 (`riingo_browse_usage`), 3

`riingo_browse_signup`
 (`riingo_browse_usage`), 3

`riingo_browse_signup()`, 14

`riingo_browse_token`
 (`riingo_browse_usage`), 3

`riingo_browse_token()`, 14

`riingo_browse_usage`, 3

`riingo_crypto_latest`, 4

`riingo_crypto_meta`, 5

`riingo_crypto_prices`, 5

`riingo_crypto_quote`, 7

`riingo_get_token` (`riingo_set_token`), 13

`riingo_iex_latest`, 8

`riingo_iex_prices`, 9

`riingo_iex_quote`, 10

`riingo_latest`, 11

`riingo_meta`, 11

`riingo_prices`, 12

`riingo_set_token`, 13

`riingo_set_token()`, 14

`supported_tickers`
 (`is_supported_ticker`), 2

`supported_tickers()`, 2