

Package ‘PoloniexR’

July 10, 2017

Title Interface to the Poloniex Cryptocurrency Trading API

Version 0.0.1

Description Provides a user-friendly R wrapper on top of the Poloniex (Cryptocurrency Trading) REST API (see <https://poloniex.com/support/api/> for more information). Results are converted into R data structures and are returned to the users in an intuitive manner. The package provides the users with two separate S4 classes:

- 'PoloniexPublicAPI': Consists of wrapper methods on top of the Poloniex Public REST API.
- 'PoloniexTradingAPI': Consists of wrapper methods on top of the Poloniex Trading REST API.

Depends R (>= 3.3)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Imports zoo (>= 1.8.0), xts (>= 0.9.7), anytime (>= 0.3.0), jsonlite (>= 1.5), httr (>= 1.2.1), digest (>= 0.6.12), methods (>= 3.3.0), RApiDatetime (>= 0.0.3),

URL <http://Essentialquant.com/blog/PoloniexR>,
<https://poloniex.com/support/api/>

BugReports <https://github.com/VermeirJellen/PoloniexR/issues>

NeedsCompilation no

Author Jellen Vermeir [aut, cre]

Maintainer Jellen Vermeir <jellenvermeir@essentialquant.com>

Repository CRAN

Date/Publication 2017-07-10 21:48:02 UTC

R topics documented:

GetPoloniexPublicCommands	2
GetPoloniexPublicCommands,PoloniexPublicAPI-method	3
GetPoloniexPublicURL	4

GetPoloniexPublicURL,PoloniexPublicAPI-method	4
GetPoloniexTradingCommands	5
GetPoloniexTradingCommands,PoloniexTradingAPI-method	5
GetPoloniexTradingURL	6
GetPoloniexTradingURL,PoloniexTradingAPI-method	7
PoloniexPublicAPI-class	7
PoloniexR	8
PoloniexTradingAPI-class	8
ProcessTradingRequest	9
ProcessTradingRequest,PoloniexTradingAPI-method	9
Return24hVolume	10
Return24hVolume,PoloniexPublicAPI-method	11
ReturnBalances	12
ReturnBalances,PoloniexTradingAPI-method	12
ReturnChartData	13
ReturnChartData,PoloniexPublicAPI-method	14
ReturnCompleteBalances	15
ReturnCompleteBalances,PoloniexTradingAPI-method	16
ReturnCurrencies	17
ReturnCurrencies,PoloniexPublicAPI-method	17
ReturnLoanOrders	18
ReturnLoanOrders,PoloniexPublicAPI-method	18
ReturnOrderBook	19
ReturnOrderBook,PoloniexPublicAPI-method	20
ReturnTicker	21
ReturnTicker,PoloniexPublicAPI-method	22
ReturnTradeHistory	23
ReturnTradeHistory,PoloniexPublicAPI-method	24
SetPoloniexPublicCommands	25
SetPoloniexPublicCommands,PoloniexPublicAPI-method	25
SetPoloniexPublicURL	26
SetPoloniexPublicURL,PoloniexPublicAPI-method	27
SetPoloniexTradingCommands	27
SetPoloniexTradingCommands,PoloniexTradingAPI-method	28
SetPoloniexTradingURL	29
SetPoloniexTradingURL,PoloniexTradingAPI-method	29

Index**31**

GetPoloniexPublicCommands

Returns REST service commands for the Poloniex Public API

Description

Returns REST service commands for the Poloniex Public API

Usage

```
GetPoloniexPublicCommands(theObject)
```

Arguments

theObject The public client API object on which the function should be called

Value

commands - a list with Key/value string pairs Represents the command strings for the respective public API functions.

Examples

```
poloniex.public <- PoloniexPublicAPI()  
GetPoloniexPublicCommands(poloniex.public)
```

GetPoloniexPublicCommands,PoloniexPublicAPI-method

Returns REST service commands for the Poloniex Public API

Description

Returns REST service commands for the Poloniex Public API

Usage

```
## S4 method for signature 'PoloniexPublicAPI'  
GetPoloniexPublicCommands(theObject)
```

Arguments

theObject The public client API object on which the function should be called

Value

commands - a list with Key/value string pairs Represents the command strings for the respective public API functions.

Examples

```
poloniex.public <- PoloniexPublicAPI()  
GetPoloniexPublicCommands(poloniex.public)
```

GetPoloniexPublicURL *Returns REST service URL for the Poloniex Public API*

Description

Returns REST service URL for the Poloniex Public API

Usage

```
GetPoloniexPublicURL(theObject)
```

Arguments

theObject The object on which the function should be called

Value

base.url - a length-one character vector. Represents the Poloniex public API service base url.

Examples

```
poloniex.public <- PoloniexPublicAPI()  
GetPoloniexPublicURL(poloniex.public)
```

GetPoloniexPublicURL,PoloniexPublicAPI-method

Returns REST service URL for the Poloniex Public API

Description

Returns REST service URL for the Poloniex Public API

Usage

```
## S4 method for signature 'PoloniexPublicAPI'  
GetPoloniexPublicURL(theObject)
```

Arguments

theObject The object on which the function should be called

Value

base.url - a length-one character vector. Represents the Poloniex public API service base url.

Examples

```
poloniex.public <- PoloniexPublicAPI()  
GetPoloniexPublicURL(poloniex.public)
```

`GetPoloniexTradingCommands`*Returns REST service commands for the Poloniex Trading API*

Description

Returns REST service commands for the Poloniex Trading API

Usage

```
GetPoloniexTradingCommands(theObject)
```

Arguments

`theObject` The Trading client API object on which the function should be called

Value

`commands` - a list with Key/value string pairs Represents the command strings for the respective Trading API functions.

Examples

```
## Not run:
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret)
GetPoloniexTradingCommands(poloniex.trading)

## End(Not run)
```

`GetPoloniexTradingCommands,PoloniexTradingAPI-method`*Returns REST service commands for the Poloniex Trading API*

Description

Returns REST service commands for the Poloniex Trading API

Usage

```
## S4 method for signature 'PoloniexTradingAPI'
GetPoloniexTradingCommands(theObject)
```

Arguments

`theObject` The Trading client API object on which the function should be called

Value

commands - a list with Key/value string pairs Represents the command strings for the respective Trading API functions.

Examples

```
## Not run:
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret)
GetPoloniexTradingCommands(poloniex.trading)

## End(Not run)
```

GetPoloniexTradingURL *Returns REST service URL for the Poloniex Trading API*

Description

Returns REST service URL for the Poloniex Trading API

Usage

```
GetPoloniexTradingURL(theObject)
```

Arguments

theObject The object on which the function should be called

Value

trading.base.url - a length-one character vector. Represents the Poloniex Trading API service base url.

Examples

```
## Not run:
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret)
GetPoloniexTradingURL(poloniex.trading)

## End(Not run)
```

 GetPoloniexTradingURL,PoloniexTradingAPI-method

Returns REST service URL for the Poloniex Trading API

Description

Returns REST service URL for the Poloniex Trading API

Usage

```
## S4 method for signature 'PoloniexTradingAPI'
GetPoloniexTradingURL(theObject)
```

Arguments

theObject The object on which the function should be called

Value

trading.base.url - a length-one character vector. Represents the Poloniex Trading API service base url.

Examples

```
## Not run:
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret)
GetPoloniexTradingURL(poloniex.trading)

## End(Not run)
```

 PoloniexPublicAPI-class

An S4 class representing the Poloniex Public API

Description

An S4 class representing the Poloniex Public API

Slots

base.url A length-one numeric vector. Represents the poloniex public API service base url.
 commands A list containing Key/Value string pairs. Represents the command strings of the associated public API functions.

 PoloniexR

R wrapper for the Poloniex Public and Trading API.

Description

This package provides a user-friendly R wrapper on top of the Poloniex (Cryptocurrency Trading) REST API. The JSON results of the API calls are translated into R Datastructures and are subsequently returned to the user in an intuitive manner.

Details

The package provides the user with two separate classes: - PoloniexPublicAPI: Allows the user to call Poloniex Public API functions. - PoloniexTradingAPI: Allows the user to call Poloniex trading API functions.

Examples: <https://github.com/VermeirJellen/PoloniexR> <http://EssentialQuant.com/blog/PoloniexR>

 PoloniexTradingAPI-class

An S4 class representing the Poloniex Trading API.

Description

All calls to the trading API are sent via HTTP POST. The POST data itself is signed with your key's secret according to the HMAC-SHA512 method.

Slots

`trading.base.url` A length-one character vector. Represents the poloniex trading API web service base url.

`key` A length-one character vector. Represents the public key of your Poloniex trading account

`secret` A length-one character vector. Represents the private key of your Poloniex trading account

`commands` A list containing Key/Value string pairs. Represents the command strings of the associated trading API functions.

ProcessTradingRequest *Process a trading request.*

Description

Process a trading request.

Usage

```
ProcessTradingRequest(theObject, command, args = list())
```

Arguments

theObject	The Trading client API object on which the function should be called.
command	- A length-one character vector. Represents the trading command that should be executed by Poloniex.
args	- list of optional trading arguments for the trading command that should be executed.

Value

List - Results of the Trading Requests. Exact contents depend on the type of Trading API Call that should be processed.

Examples

```
## Not run:
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret)
deposit.addresses <- ProcessTradingRequest(poloniex.trading,
  command = poloniex.trading@commands$returnDepositAddresses)

## End(Not run)
```

ProcessTradingRequest, PoloniexTradingAPI-method
Process a trading request.

Description

Process a trading request.

Usage

```
## S4 method for signature 'PoloniexTradingAPI'
ProcessTradingRequest(theObject, command,
  args = list())
```

Arguments

theObject	The Trading client API object on which the function should be called.
command	- A length-one character vector. Represents the trading command that should be executed by Poloniex.
args	- list of optional trading arguments for the trading command that should be executed.

Value

List - Results of the Trading Requests. Exact contents depend on the type of Trading API Call that should be processed.

Examples

```
## Not run:
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret)
deposit.addresses <- ProcessTradingRequest(poloniex.trading,
  command = poloniex.trading@commands$returnDepositAddresses)

## End(Not run)
```

Return24hVolume	<i>Returns 24-hour trading volumes for all available markets. Additionally, total trading volumes for the 5 primary currencies are also retrieved.</i>
-----------------	--

Description

Returns 24-hour trading volumes for all available markets. Additionally, total trading volumes for the 5 primary currencies are also retrieved.

Usage

```
Return24hVolume(theObject)
```

Arguments

theObject	The public client API object on which the function should be called
-----------	---

Value

A list containing the 24 hour volume information. volume.pairs: A dataframe Rows - Cryptocurrency Trading Pairs. Every pair represents a market. Columns - Transacted trading volumes for a particular market. Volumes are expressed in both base currency units and quoted currency units. volume.totals: Total trading volumes of five primary currencies.

Examples

```
poloniex.public <- PoloniexPublicAPI()
volume.info <- Return24hVolume(poloniex.public)
head(volume.info$volume.pairs)
volume.info$volume.totals
```

Return24hVolume,PoloniexPublicAPI-method

Returns 24-hour trading volumes for all available markets. Additionally, total trading volumes for the 5 primary currencies are also retrieved.

Description

Returns 24-hour trading volumes for all available markets. Additionally, total trading volumes for the 5 primary currencies are also retrieved.

Usage

```
## S4 method for signature 'PoloniexPublicAPI'
Return24hVolume(theObject)
```

Arguments

theObject The public client API object on which the function should be called

Value

A list containing the 24 hour volume information. volume.pairs: A dataframe Rows - Cryptocurrency Trading Pairs. Every pair represents a market. Columns - Transacted trading volumes for a particular market. Volumes are expressed in both base currency units and quoted currency units. volume.totals: Total trading volumes of five primary currencies.

Examples

```
poloniex.public <- PoloniexPublicAPI()
volume.info <- Return24hVolume(poloniex.public)
head(volume.info$volume.pairs)
volume.info$volume.totals
```

ReturnBalances	<i>Returns all the available non-zero balances associated with this account.</i>
----------------	--

Description

Returns all the available non-zero balances associated with this account.

Usage

```
ReturnBalances(theObject)
```

Arguments

theObject The Trading client API object on which the function should be called.

Value

a vector containing non-zero currency balances. Each entry corresponds to a balance for a single currency.

Examples

```
## Not run:
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret)
balances        <- ReturnBalances(poloniex.trading)
head(balances)

## End(Not run)
```

ReturnBalances,PoloniexTradingAPI-method	<i>Returns all the available non-zero balances associated with this account.</i>
--	--

Description

Returns all the available non-zero balances associated with this account.

Usage

```
## S4 method for signature 'PoloniexTradingAPI'
ReturnBalances(theObject)
```

Arguments

theObject The Trading client API object on which the function should be called.

Value

a vector containing non-zero currency balances. Each entry corresponds to a balance for a single currency.

Examples

```
## Not run:
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret)
balances <- ReturnBalances(poloniex.trading)
head(balances)

## End(Not run)
```

ReturnChartData	<i>Fetch OHLC and volume timeseries data for a specific market. The caller must specify a date range interval and datafrequency for the requested timeseries.</i>
-----------------	---

Description

Fetch OHLC and volume timeseries data for a specific market. The caller must specify a date range interval and datafrequency for the requested timeseries.

Usage

```
ReturnChartData(theObject, pair = "USDT_BTC",
  from = as.POSIXct("2014-01-01 00:00:00 UTC"),
  to = as.POSIXct("2018-01-01 00:00:00 UTC"), period = "4H")
```

Arguments

theObject	The object on which the function should be called.
pair	a length-one character vector - The currencypair for which timeseries information should be fetched.
from	POSIXct - Starting timestamp for daterange interval.
to	POSIXct - Ending timestamp for daterange interval.
period	length-one character vector OR length-one numeric vector - datafrequency for requested timeseries. if is(period, "numeric"): 300 / 900 / 1800 / 7200 / 14400 / 86400 if is(period, "character"): "5M", "15M", "30M", "2H", "4H", "D"

Value

an xts timeseries object. rows - ohcl and volume information for one timestamp. columns - high, low, open, close, volume, quotevolume, weightedaverage.

Examples

```

Sys.setenv(tz = "UTC")
poloniex.public <- PoloniexPublicAPI()
pair <- "BTC_NXT"
from <- as.POSIXct("2012-04-01 00:00:00 UTC")
to <- as.POSIXct("2012-04-09 00:00:00 UTC")
period <- "4H"

chart.data <- ReturnChartData(theObject = poloniex.public,
                             pair = pair,
                             from = from,
                             to = to,
                             period = period)

```

ReturnChartData,PoloniexPublicAPI-method

Fetch OHLC and volume timeseries data for a specific market. The caller must specify a date range interval and datafrequency for the requested timeseries.

Description

Fetch OHLC and volume timeseries data for a specific market. The caller must specify a date range interval and datafrequency for the requested timeseries.

Usage

```

## S4 method for signature 'PoloniexPublicAPI'
ReturnChartData(theObject, pair = "USDT_BTC",
               from = as.POSIXct("2014-01-01 00:00:00 UTC"),
               to = as.POSIXct("2018-01-01 00:00:00 UTC"), period = "4H")

```

Arguments

theObject	The object on which the function should be called.
pair	a length-one character vector - The currency pair for which timeseries information should be fetched.
from	POSIXct - Starting timestamp for daterange interval.
to	POSIXct - Ending timestamp for daterange interval.
period	length-one character vector OR length-one numeric vector - datafrequency for requested timeseries. if is(period, "numeric"): 300 / 900 / 1800 / 7200 / 14400 / 86400 if is(period, "character"): "5M", "15M", "30M", "2H", "4H", "D"

Value

an xts timeseries object. rows - ohcl and volume information for one timestamp. columns - high, low, open, close, volume, quotevolume, weightedaverage.

Examples

```

Sys.setenv(tz = "UTC")
poloniex.public <- PoloniexPublicAPI()
pair <- "BTC_NXT"
from <- as.POSIXct("2010-01-01 00:00:00 UTC")
to <- as.POSIXct("2012-04-09 00:00:00 UTC")
period <- "4H"

chart.data <- ReturnChartData(theObject = poloniex.public,
                             pair = pair,
                             from = from,
                             to = to,
                             period = period)

```

ReturnCompleteBalances

Returns all of your account balances, including available balance, balance on orders, and the estimated BTC value of your balance. By default, this call is limited to your exchange account; set the all.balances input parameter to TRUE to include your margin and lending accounts.

Description

Returns all of your account balances, including available balance, balance on orders, and the estimated BTC value of your balance. By default, this call is limited to your exchange account; set the all.balances input parameter to TRUE to include your margin and lending accounts.

Usage

```
ReturnCompleteBalances(theObject, all.balances = FALSE)
```

Arguments

theObject The Trading client API object on which the function should be called.
all.balances logical - set to TRUE to also return margin and lending account balances.

Value

todo

Examples

```

## Not run:
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret)
balances <- ReturnCompleteBalances(poloniex.trading)
head(balances)

```

```
balances.all <- ReturnCompleteBalances(poloniex.trading)
head(balances.all)

## End(Not run)
```

ReturnCompleteBalances,PoloniexTradingAPI-method

Returns all of your account balances, including available balance, balance on orders, and the estimated BTC value of your balance. By default, this call is limited to your exchange account; set the all.balances input parameter to TRUE to include your margin and lending accounts.

Description

Returns all of your account balances, including available balance, balance on orders, and the estimated BTC value of your balance. By default, this call is limited to your exchange account; set the all.balances input parameter to TRUE to include your margin and lending accounts.

Usage

```
## S4 method for signature 'PoloniexTradingAPI'
ReturnCompleteBalances(theObject,
  all.balances = FALSE)
```

Arguments

theObject The Trading client API object on which the function should be called.
all.balances logical - set to TRUE to also return margin and lending account balances.

Value

todo

Examples

```
## Not run:
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret)
balances         <- ReturnCompleteBalances(poloniex.trading)
head(balances)

balances.all <- ReturnCompleteBalances(poloniex.trading)
head(balances.all)

## End(Not run)
```

ReturnCurrencies	<i>Returns information about currencies</i>
------------------	---

Description

Returns information about currencies

Usage

```
ReturnCurrencies(theObject)
```

Arguments

theObject The object on which the function should be called

Value

A dataframe with currency information - rows: Information about a single currency - columns: id, txfee, minConf, disabled, delisted, frozen

Examples

```
poloniex.public <- PoloniexPublicAPI()
currencies <- ReturnCurrencies(poloniex.public)
```

ReturnCurrencies,PoloniexPublicAPI-method	<i>Returns information about currencies</i>
---	---

Description

Returns information about currencies

Usage

```
## S4 method for signature 'PoloniexPublicAPI'
ReturnCurrencies(theObject)
```

Arguments

theObject The object on which the function should be called

Value

A dataframe with currency information - rows: Information about a single currency - columns: id, txfee, minConf, disabled, delisted, frozen

Examples

```
poloniex.public <- PoloniexPublicAPI()
currencies <- ReturnCurrencies(poloniex.public)
```

ReturnLoanOrders	<i>Returns a list of loan offers and demands for a given currency.</i>
------------------	--

Description

Returns a list of loan offers and demands for a given currency.

Usage

```
ReturnLoanOrders(theObject, currency = "BTC")
```

Arguments

theObject	The object on which the function should be called
currency	- a length-one character vector Represents the currency for which loan information should be fetched.

Value

A list containing two items. Each item is a dataframe containing the actual loan offers or demands information. eg list\$offers / list\$demands - rows: offers / demand "order book" entries. - columns: rate, amount, min.days, max.days

Examples

```
poloniex.public <- PoloniexPublicAPI()
currency <- "BTC"
loan.orders <- ReturnLoanOrders(poloniex.public)
```

ReturnLoanOrders,PoloniexPublicAPI-method	<i>Returns a list of loan offers and demands for a given currency.</i>
---	--

Description

Returns a list of loan offers and demands for a given currency.

Usage

```
## S4 method for signature 'PoloniexPublicAPI'
ReturnLoanOrders(theObject, currency = "BTC")
```

Arguments

theObject The object on which the function should be called

currency - a length-one character vector Represents the currency for which loan information should be fetched.

Value

A list containing two items. Each item is a dataframe containing the actual loan offers or demands information. eg list\$offers / list\$demands - rows: offers / demand "order book" entries. - columns: rate, amount, min.days, max.days

Examples

```
poloniex.public <- PoloniexPublicAPI()
currency        <- "BTC"
loan.orders     <- ReturnLoanOrders(poloniex.public)
```

ReturnOrderBook	<i>Returns the order book for a given market, as well as a sequence number for use with the Push API and an indicator specifying whether the market is frozen.</i>
-----------------	--

Description

Returns the order book for a given market, as well as a sequence number for use with the Push API and an indicator specifying whether the market is frozen.

Usage

```
ReturnOrderBook(theObject, pair = "all", depth = 10)
```

Arguments

theObject The public client API object on which the function should be called.

pair length one-character vector - The currencypair for which orderbook information should be fetched. You may set pair to "all" to fetch the order books of all markets.

depth numeric - depth of the orderbook.

Value

A list containing orderbook information.

if pair == "all": a list containing orderbook information for all available markets. Each list entry contains information for one specific market. if !pair == "all": a list containing orderbook information for the requested markets.

Each market list contains following fields: - ask: Orderbook sell side, Dataframe containing ask prices and corresponding amounts. - bid: Orderbook buy side. Dataframe containing bid prices and corresponding amounts. - frozen: indicator specifying wheather market is frozen or not. - seq: Sequence number for Push API.

Examples

```
poloniex.public <- PoloniexPublicAPI()

pair <- "BTC_NXT"
depth <- 100
order.book <- ReturnOrderBook(poloniex.public,
                              pair = pair,
                              depth = 10)

order.book$bid
order.book$ask
order.book$frozen
order.book$seq

pair <- "all"
depth <- 10
order.book <- ReturnOrderBook(poloniex.public,
                              pair = pair,
                              depth = 10)

names(order.book)
order.book$BTC_ETH$ask
```

ReturnOrderBook,PoloniexPublicAPI-method

Returns the order book for a given market, as well as a sequence number for use with the Push API and an indicator specifying whether the market is frozen.

Description

Returns the order book for a given market, as well as a sequence number for use with the Push API and an indicator specifying whether the market is frozen.

Usage

```
## S4 method for signature 'PoloniexPublicAPI'
ReturnOrderBook(theObject, pair = "all",
                depth = 10)
```

Arguments

theObject	The public client API object on which the function should be called.
pair	length one-character vector - The currencypair for which orderbook information should be fetched. You may set pair to "all" to fetch the order books of all markets.
depth	numeric - depth of the orderbook.

Value

A list containing orderbook information.

if pair == "all": a list containing orderbook information for all available markets. Each list entry contains information for one specific market. if !pair == "all": a list containing orderbook information for the requested markets.

Each market list contains following fields: - ask: Orderbook sell side, Dataframe containing ask prices and corresponding amounts. - bid: Orderbook buy side. Dataframe containing bid prices and corresponding amounts. - frozen: indicator specifying wheather market is frozen or not. - seq: Sequence number for Push API.

Examples

```
poloniex.public <- PoloniexPublicAPI()

pair <- "BTC_NXT"
depth <- 2
order.book <- ReturnOrderBook(poloniex.public,
                              pair = pair,
                              depth = depth)

order.book$bid
order.book$ask
order.book$frozen
order.book$seq

pair <- "all"
depth <- 2
order.book <- ReturnOrderBook(poloniex.public,
                              pair = pair,
                              depth = depth)

names(order.book)
order.book$BTC_ETH$ask
```

ReturnTicker

Returns the ticker for all markets

Description

Returns the ticker for all markets

Usage

```
ReturnTicker(theObject)
```

Arguments

theObject The public client API object on which the function should be called

Value

A dataframe containing ticker information Rows - Cryptocurrency Pairs Columns - id, last, lowest-Ask, highestBid, percentChange baseVolume, quoteVolume, isFrozen, high24hr, low24hr

Examples

```
poloniex.public <- PoloniexPublicAPI()
GetPoloniexPublicURL(poloniex.public)
```

ReturnTicker,PoloniexPublicAPI-method
Returns the ticker for all markets

Description

Returns the ticker for all markets

Usage

```
## S4 method for signature 'PoloniexPublicAPI'
ReturnTicker(theObject)
```

Arguments

theObject The public client API object on which the function should be called

Value

A dataframe containing ticker information Rows - Cryptocurrency Pairs Columns - id, last, lowest-Ask, highestBid, percentChange baseVolume, quoteVolume, isFrozen, high24hr, low24hr

Examples

```
poloniex.public <- PoloniexPublicAPI()
GetPoloniexPublicURL(poloniex.public)
```

ReturnTradeHistory	<i>Returns the past 200 trades for a given market, or up to 50,000 trades inside a specified input date range interval.</i>
--------------------	---

Description

Returns the past 200 trades for a given market, or up to 50,000 trades inside a specified input date range interval.

Usage

```
ReturnTradeHistory(theObject, pair = "USDT_BTC", from = NULL, to = NULL)
```

Arguments

theObject	The public client API object on which the function should be called.
pair	length-one character vector - The currency pair for which trade history information should be fetched.
from	POSIXct (or NULL) - starting timestamp for optional date range interval. if from != NULL: from must be < 'to' & from must be >= (to - 1 year)
to	POSIXct (or NULL) - ending timestamp for optional date range interval.

Value

an xts object containing historical trade information

Rows - Information for one specific trade. Columns - globalTradeID, tradeID, type, rate, amount, total

if from == NULL: Receive information for last 200 trades before 'to' if to == NULL: Ending timestamp for daterange will be set to Sys.time()

Examples

```

Sys.setenv(tz = "UTC")
poloniex.public <- PoloniexPublicAPI()
pair <- "BTC_NXT"
from <- as.POSIXct("2017-02-01 00:00:00 UTC")
to <- as.POSIXct("2017-02-02 00:00:00 UTC")

trades.data <- ReturnTradeHistory(theObject = poloniex.public,
                                  pair      = pair,
                                  from      = from,
                                  to       = to)

```

ReturnTradeHistory,PoloniexPublicAPI-method

Returns the past 200 trades for a given market, or up to 50,000 trades inside a specified input date range interval.

Description

Returns the past 200 trades for a given market, or up to 50,000 trades inside a specified input date range interval.

Usage

```
## S4 method for signature 'PoloniexPublicAPI'
ReturnTradeHistory(theObject, pair = "USDT_BTC",
  from = NULL, to = NULL)
```

Arguments

theObject	The public client API object on which the function should be called.
pair	length-one character vector - The currencypair for which trade history information should be fetched.
from	POSIXct (or NULL) - starting timestamp for optional date range interval. if from != NULL: from must be < 'to' & from must be >= (to - 1 year)
to	POSIXct (or NULL) - ending timestamp for optional date range interval.

Value

an xts object containing historical trade information

Rows - Information for one specific trade. Columns - globalTradeID, tradeID, type, rate, amount, total

if from == NULL: Receive information for last 200 trades before 'to' if to == NULL: Ending timestamp for daterange will be set to Sys.time()

Examples

```
Sys.setenv(tz = "UTC")
poloniex.public <- PoloniexPublicAPI()
pair <- "BTC_NXT"
from <- as.POSIXct("2017-07-04 00:00:00 UTC")
to <- as.POSIXct("2017-07-05 00:00:00 UTC")

trades.data <- ReturnTradeHistory(theObject = poloniex.public,
  pair = pair,
  from = from,
  to = to)
```

 SetPoloniexPublicCommands

Setter for REST service commands for the Poloniex public API.

Description

Setter for REST service commands for the Poloniex public API.

Usage

```
SetPoloniexPublicCommands(theObject, commands)
```

Arguments

theObject	The public client API object on which the function should be called
commands	- a list with Key/value string pairs Represents the command strings for the respective public API functions.

Value

theObject - a copy of the object with the modified command slot

Examples

```
poloniex.public <- PoloniexPublicAPI()
GetPoloniexPublicCommands(poloniex.public)
commands.new <- list(returnTicker = "returnTicker.new",
                    return24hVolume = "return24hVolume.new")
poloniex.public <- SetPoloniexPublicCommands(poloniex.public,
                                           commands = commands.new)
GetPoloniexPublicCommands(poloniex.public)
```

 SetPoloniexPublicCommands, PoloniexPublicAPI-method

Setter for REST service commands for the Poloniex public API.

Description

Setter for REST service commands for the Poloniex public API.

Usage

```
## S4 method for signature 'PoloniexPublicAPI'
SetPoloniexPublicCommands(theObject, commands)
```

Arguments

theObject The public client API object on which the function should be called

commands - a list with Key/value string pairs Represents the command strings for the respective public API functions.

Value

theObject - a copy of the object with the modified command slot

Examples

```
poloniex.public <- PoloniexPublicAPI()
GetPoloniexPublicCommands(poloniex.public)
commands.new <- list(returnTicker = "returnTicker.new",
                    return24hVolume = "return24hVolume.new")
poloniex.public <- SetPoloniexPublicCommands(poloniex.public,
                                           commands = commands.new)
GetPoloniexPublicCommands(poloniex.public)
```

SetPoloniexPublicURL *Setter for REST service URL for the Poloniex public API*

Description

Setter for REST service URL for the Poloniex public API

Usage

```
SetPoloniexPublicURL(theObject, base.url)
```

Arguments

theObject The public client API object on which the function should be called

base.url length-one character vector. Represents the REST service URL.

Value

theObject - a copy of the object with the modified base.url slot

Examples

```
poloniex.public <- PoloniexPublicAPI(base.url = "https://not_working/public?")
poloniex.public <- SetPoloniexPublicURL(poloniex.public,
                                       base.url = "https://poloniex.com/public?")
GetPoloniexPublicURL(poloniex.public)
```

SetPoloniexPublicURL,PoloniexPublicAPI-method

Setter for REST service URL for the Poloniex public API

Description

Setter for REST service URL for the Poloniex public API

Usage

```
## S4 method for signature 'PoloniexPublicAPI'  
SetPoloniexPublicURL(theObject, base.url)
```

Arguments

theObject The public client API object on which the function should be called
base.url length-one character vector. Represents the REST service URL.

Value

theObject - a copy of the object with the modified base.url slot

Examples

```
poloniex.public <- PoloniexPublicAPI(base.url = "https://not_working/public?")  
poloniex.public <- SetPoloniexPublicURL(poloniex.public,  
                                       base.url = "https://poloniex.com/public?")  
GetPoloniexPublicURL(poloniex.public)
```

SetPoloniexTradingCommands

Setter for REST service commands for the Poloniex Trading API.

Description

Setter for REST service commands for the Poloniex Trading API.

Usage

```
SetPoloniexTradingCommands(theObject, commands)
```

Arguments

theObject The Trading client API object on which the function should be called
commands - a list with Key/value string pairs Represents the command strings for the re-
 spective Trading API functions.


```

GetPoloniexTradingCommands(poloniex.trading)

## End(Not run)

```

SetPoloniexTradingURL *Setter for REST service URL for the Poloniex Trading API*

Description

Setter for REST service URL for the Poloniex Trading API

Usage

```
SetPoloniexTradingURL(theObject, trading.base.url)
```

Arguments

theObject The Trading client API object on which the function should be called
trading.base.url length-one character vector. Represents the REST service URL.

Value

theObject - a copy of the object with the modified trading.base.url slot

Examples

```

## Not run:
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret,
                                     trading.base.url = "https://not_working/trading?")
poloniex.trading <- SetPoloniexTradingURL(poloniex.trading,
                                     trading.base.url = "https://poloniex.com/tradingApi?")
GetPoloniexTradingURL(poloniex.trading)

## End(Not run)

```

SetPoloniexTradingURL,PoloniexTradingAPI-method
Setter for REST service URL for the Poloniex Trading API

Description

Setter for REST service URL for the Poloniex Trading API

Usage

```
## S4 method for signature 'PoloniexTradingAPI'  
SetPoloniexTradingURL(theObject,  
  trading.base.url)
```

Arguments

theObject The Trading client API object on which the function should be called
trading.base.url
 length-one character vector. Represents the REST service URL.

Value

theObject - a copy of the object with the modified trading.base.url slot

Examples

```
## Not run:  
poloniex.trading <- PoloniexTradingAPI(your.key, your.secret,  
  trading.base.url = "https://not_working/trading?")  
poloniex.trading <- SetPoloniexTradingURL(poloniex.trading,  
  trading.base.url = "https://poloniex.com/tradingApi?")  
GetPoloniexTradingURL(poloniex.trading)  
  
## End(Not run)
```

Index

GetPoloniexPublicCommands, [2](#)
GetPoloniexPublicCommands, PoloniexPublicAPI-method, [3](#)
GetPoloniexPublicURL, [4](#)
GetPoloniexPublicURL, PoloniexPublicAPI-method, [4](#)
GetPoloniexTradingCommands, [5](#)
GetPoloniexTradingCommands, PoloniexTradingAPI-method, [5](#)
GetPoloniexTradingURL, [6](#)
GetPoloniexTradingURL, PoloniexTradingAPI-method, [7](#)

PoloniexPublicAPI
 (PoloniexPublicAPI-class), [7](#)
PoloniexPublicAPI-class, [7](#)
PoloniexR, [8](#)
PoloniexR-package (PoloniexR), [8](#)
PoloniexTradingAPI
 (PoloniexTradingAPI-class), [8](#)
PoloniexTradingAPI-class, [8](#)
ProcessTradingRequest, [9](#)
ProcessTradingRequest, PoloniexTradingAPI-method, [9](#)

Return24hVolume, [10](#)
Return24hVolume, PoloniexPublicAPI-method, [11](#)
ReturnBalances, [12](#)
ReturnBalances, PoloniexTradingAPI-method, [12](#)
ReturnChartData, [13](#)
ReturnChartData, PoloniexPublicAPI-method, [14](#)
ReturnCompleteBalances, [15](#)
ReturnCompleteBalances, PoloniexTradingAPI-method, [16](#)
ReturnCurrencies, [17](#)
ReturnCurrencies, PoloniexPublicAPI-method, [17](#)

ReturnLoanOrders, [18](#)
ReturnLoanOrders, PoloniexPublicAPI-method, [18](#)
ReturnOrderBook, [19](#)
ReturnOrderBook, PoloniexPublicAPI-method, [20](#)
ReturnTicker, [21](#)
ReturnTicker, PoloniexPublicAPI-method, [22](#)
ReturnTradeHistory, [23](#)
ReturnTradeHistory, PoloniexPublicAPI-method, [24](#)

SetPoloniexPublicCommands, [25](#)
SetPoloniexPublicCommands, PoloniexPublicAPI-method, [25](#)
SetPoloniexPublicURL, [26](#)
SetPoloniexPublicURL, PoloniexPublicAPI-method, [27](#)
SetPoloniexTradingCommands, [27](#)
SetPoloniexTradingCommands, PoloniexTradingAPI-method, [28](#)
SetPoloniexTradingURL, [29](#)
SetPoloniexTradingURL, PoloniexTradingAPI-method, [29](#)