Package 'KrakenR'

<p style="text-align:center">October 22, 2024</p>

Title Comprehensive R Interface for Accessing Kraken Cryptocurrency
Exchange REST API

Version 1.0.0

Description A comprehensive R interface to access data from the Kraken cryptocurrency exchange REST API <https://docs.kraken.com/api/>.
It allows users to retrieve various market data, such as asset information, trading pairs, and price data.
The package is designed to facilitate efficient data access for analysis, strategy development, and monitoring of cryptocurrency market trends.

URL https://github.com/nathanael-g-durst/KrakenR

BugReports https://github.com/nathanael-g-durst/KrakenR/issues

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports jsonlite, dplyr, tidyr, anytime, magrittr, rlang

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Nathanaël Dürst [aut, cre]

Maintainer Nathanaël Dürst <nathanael.durst@unil.ch>

Repository CRAN

Date/Publication 2024-10-22 14:00:05 UTC

Contents

---

getAssets   *Retrieve Asset Information from Kraken Exchange*

Description

This function fetches detailed asset information from the Kraken API, either for all available assets or a specified subset.

Usage

```
getAssets(assets = "All")
```

Arguments

assets   A character vector specifying the assets to retrieve. Use "All" to fetch data for all
                assets. For specific assets, provide their abbreviations (e.g., "ADA" or c("BTC",
                "ETH")). Default is "All".

Value

A data frame containing detailed information about the requested assets.

Examples

```
getAssets()
getAssets("ETH")
getAssets(c("BTC", "AAVE", "ADA"))
```

---

getOB   *Retrieve Order Book Data from Kraken Exchange*

Description

This function fetches order book data from the Kraken API for a specified trading pair.

Usage

```
getOB(pair, count = NULL)
```

Arguments

pair	A character string specifying the trading pair (e.g., "ADAEUR"). This is a required parameter.
count	An optional integer between 1 and 500 specifying the number of orders to retrieve. Default is NULL.

Value

A data frame containing the order book data for the requested trading pair, with bid orders appearing first (sorted by price ascending) followed by ask orders (sorted by price ascending).

Examples

```
getOB("ADAEUR")
getOB("ADAEUR", count = 100)
```

---

getOHLC   *Retrieve OHLC Data from Kraken Exchange*

Description

This function fetches OHLC (Open, High, Low, Close) data from the Kraken API for a specified trading pair.

Usage

```
getOHLC(pair, interval = 1, since = NULL)
```

Arguments

pair	A character string specifying the trading pair (e.g., "ADAEUR"). This is a required parameter.
interval	A time frame interval in minutes or in a user-friendly format (e.g., "1h" for 1 hour, "1d" for 1 day, etc.). Possible values are "1m", "5m", "15m", "30m", "1h", "4h", "1d", "1w", "2w", or numeric values in minutes. Default is 1 minute.
since	A character string for a human-readable date-time (e.g., "2024-10-01 12:00:00") or a Unix timestamp. Default is NULL (returns all available data).

Value

A data frame containing OHLC data for the requested trading pair.

Examples

```
getOHLC("ADAEUR")
getOHLC("ADAEUR", interval = "4h")
getOHLC("ADAEUR", interval = 1440, since = "2024-01-01 00:00:00")
getOHLC("ADAEUR", interval = "2w", since = 1704063600)
```

---

getPairs	*Retrieve Tradable Asset Pairs Information from Kraken Exchange*

Description

This function fetches detailed information about tradable asset pairs from the Kraken API, either for all pairs or a specified subset.

Usage

```
getPairs(pairs = "All", info = "info", country_code = NULL)
```

Arguments

pairs	A character vector specifying the pairs to retrieve. Use "All" to fetch data for all pairs. For specific pairs, provide their abbreviations (e.g., "ADAEUR" or c("ADAEUR", "BTCUSD")). Default is "All".
info	A character vector to specify the level of detail ("info", "leverage", "fees", "margin"). Default is "info".
country_code	A single character string specifying the country code to filter the pairs. Use NULL if not needed. Default is NULL.

Value

A data frame containing detailed information about the requested asset pairs with nested lists for columns with multiple values.

Examples

```
getPairs()
getPairs("ADAEUR")
getPairs(c("ADAEUR", "BTCUSD"), info = "fees")
getPairs("ADAEUR", country_code = "CH")
getPairs(country_code = "US:TX")
```

---

getSpreads	*Retrieve Recent Spreads Data from Kraken Exchange*

Description

This function fetches recent spread data from the Kraken API for a specified trading pair.

Usage

```
getSpreads(pair, since = NULL, timestamp = FALSE)
```

Arguments

pair	A character string specifying the trading pair (e.g., "XTZUSD", "ADAEUR"). This is a required parameter.
since	A character string for a human-readable date-time (e.g., "2024-10-01 12:00:00") or a Unix timestamp. Default is NULL.
timestamp	A logical value. If TRUE, the function returns both the spreads data frame and the last timestamp for future polling. If FALSE (default), it returns only the data frame.

Value

A data frame containing the spread data or a list containing the data frame and the last timestamp.

Examples

```
getSpreads("XTZUSD")
getSpreads("ADAEUR", since = "2024-10-01 00:00:00", timestamp = TRUE)
```

---

getStatus   *Retrieve the Current System Status of the Kraken Exchange*

Description

This function fetches the current system status from the Kraken API, including both the operational status and the timestamp.

Usage

```
getStatus(data = "both")
```

Arguments

data   A character string specifying the type of data to return. Use "status" for the system status, "timestamp" for the timestamp, or "both" to retrieve both status and timestamp. The default is "both".

Value

A character string if data = "status" or data = "timestamp". A named vector if data = "both".

Examples

```
getStatus("both")
getStatus("status")
getStatus("timestamp")
```

---

getTickers   *Retrieve Ticker Information from Kraken Exchange*

Description

This function fetches detailed ticker information from the Kraken API for specified trading pairs or all available pairs.

Usage

```
getTickers(pairs = "All")
```

Arguments

pairs   A character vector specifying the pairs to retrieve ticker information for. Use "All" to fetch data for all pairs. For specific pairs, provide their abbreviations (e.g., "ADAEUR" or c("ADAEUR", "BTCUSD")). Default is "All".

Value

A data frame containing detailed ticker information for the requested pairs with renamed and split columns, where applicable, and all numbers treated as numeric.

Examples

```
getTickers()
getTickers("ADAEUR")
getTickers(c("ADAEUR", "BTCUSD"))
```

---

getTime   *Retrieve the Current Kraken Server Time*

Description

This function fetches the current server time from the Kraken API in either UNIX timestamp or RFC 1123 format.

Usage

```
getTime(format = "unix")
```

Arguments

format   A character string specifying the time format to return. Use "unix" for UNIX
                  timestamp or "rfc" for RFC 1123 format. The default is "unix".

Value

A POSIXct object if format = "unix" or a character string in RFC 1123 format if format = "rfc".

Examples

```
getTime("unix")
getTime("rfc")
```

getTrades	*Retrieve Recent Trades Data from Kraken Exchange*

Description

This function fetches recent trade data from the Kraken API for a specified trading pair.

Usage

```
getTrades(pair, since = NULL, count = NULL)
```

Arguments

pair   A character string specifying the trading pair (e.g., "XTZUSD", "ADAEUR").
              This is a required parameter.

since   A character string for a human-readable date-time (e.g., "2024-10-01 12:00:00")
              or a Unix timestamp. Default is NULL (returns all available trades).

count   An optional integer between 1 and 1000 specifying the number of trades to re-
              trieve. Default is NULL (returns up to 1000 trades).

Value

A data frame containing the recent trade data for the requested trading pair.

Examples

```
getTrades("XTZUSD")
getTrades("XTZUSD", since = "2024-10-01")
getTrades("XTZUSD", since = "2024-10-01 00:00:00", count = 100)
```

Index