

# Package: td (via r-universe)

August 29, 2024

**Type** Package

**Title** Access to the 'twelvedata' Financial Data API

**Version** 0.0.6.2

**Date** 2022-01-26

**Author** Dirk Eddelbuettel and Kenneth Rose

**Maintainer** Dirk Eddelbuettel <edd@debian.org>

**Description** The 'twelvedata' REST service offers access to current and historical data on stocks, standard as well as digital 'crypto' currencies, and other financial assets covering a wide variety of course and time spans. See <<https://twelvedata.com/>> for details, to create an account, and to request an API key for free-but-capped access to the data.

**License** GPL (>= 2)

**Imports** RcppSimdJson

**Suggests** tinytest, xts

**URL** <https://dirk.eddelbuettel.com/code/td.html>,  
<https://github.com/eddelbuettel/td>

**BugReports** <https://github.com/eddelbuettel/td/issues>

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Repository** <https://eddelbuettel.r-universe.dev>

**RemoteUrl** <https://github.com/eddelbuettel/td>

**RemoteRef** HEAD

**RemoteSha** b69255e926acbdfe53598b26f3ab6fcb52b29447

## Contents

td-package . . . . .	2
fun_profile . . . . .	3

fun_statistics . . . . .	3
get_price . . . . .	4
get_quote . . . . .	6
ref_crypto_exchanges . . . . .	7
ref_crypto_pairs . . . . .	8
ref_earliest_timestamp . . . . .	9
ref_etf . . . . .	10
ref_exchanges . . . . .	11
ref_forex_pairs . . . . .	12
ref_indices . . . . .	13
ref_stocks . . . . .	13
ref_symbol_search . . . . .	15
store_key . . . . .	15
time_series . . . . .	16
<b>Index</b>	<b>20</b>

---

td-package	<i>Access to the 'twelvedata' Financial Data API</i>
------------	------------------------------------------------------

---

**Description**

The 'twelvedata' REST service offers access to current and historical data on stocks, standard as well as digital 'crypto' currencies, and other financial assets covering a wide variety of course and time spans. See <<https://twelvedata.com/>> for details, to create an account, and to request an API key for free-but-capped access to the data.

**Package Content**

Index: This package was not yet installed at build time.

**Maintainer**

Dirk Eddelbuettel <[edd@debian.org](mailto:edd@debian.org)>

**Author(s)**

Dirk Eddelbuettel and Kenneth Rose

---

fun_profile	<i>Fundamental Data Accessor for the Profile from 'twelvedata' This API endpoint is available starting with the Grow plan.</i>
-------------	--------------------------------------------------------------------------------------------------------------------------------

---

**Description**

Retrieve Profile Data from 'twelvedata'.

**Usage**

```
fun_profile(sym, exchange = "", country = "", apikey)
```

**Arguments**

sym	(character) A (single or vector) symbol for Indices. Defaults to unset.
exchange	(optional, character) Exchange. Defaults to unset.
country	(optional, character) An alpha code or country name. Defaults to unset.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by tools::R_user_dir (for R 4.0.0 or later), or the TWELVEDATA_API_KEY variable.

**Details**

fun\_profile.

**Value**

The requested data is returned as a list object.

**See Also**

<https://twelvedata.com/docs>

---

fun_statistics	<i>Fundamental Data Accessor for Main Statistics from 'twelvedata' This API endpoint is available starting with the Pro plan.</i>
----------------	-----------------------------------------------------------------------------------------------------------------------------------

---

**Description**

Retrieve Main Fundamental Data Statistics Including Valuation Metrics and Financials from 'twelvedata'.

**Usage**

```
fun_statistics(sym, exchange = "", country = "", apikey)
```

**Arguments**

sym	(character) A (single or vector) symbol for Indices. Defaults to unset.
exchange	(optional, character) Exchange. Defaults to unset.
country	(optional, character) An alpha code or country name. Defaults to unset.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by <code>tools::R_user_dir</code> (for R 4.0.0 or later), or the <code>TWELVEDATA_API_KEY</code> variable.

**Details**

`fun_statistics.`

**Value**

The requested data is returned as a list object.

**See Also**

<https://twelvedata.com/docs>

---

get\_price

*Quote Data Accessor for 'twelvedata'*

---

**Description**

Retrieve Securities Real-Time Price Data from 'twelvedata'

**Usage**

```
get_price(
  sym,
  as = c("data.frame", "raw"),
  exchange = "",
  country = "",
  type = c(NA_character_, "Stock", "Index", "ETF", "REIT"),
  dp = 5,
  apikey
)
```

**Arguments**

sym	(character) A (single or vector) symbol understood by the backend as a stock symbol, foreign exchange pair, or more. See the ‘twelvedata’ documentation for details on what is covered. In the case of a vector of arguments a vector of prices is returned.
as	(optional, character) A selector for the desired output format: one of “data.frame” (the default) or “raw”.
exchange	(optional, character) A selection of the exchange for which data for “sym” is requested, default value is unset.
country	(optional, character) A selection of the country exchange for which data for “sym” is requested, default value is unset.
type	(optional, character) A valid security type selection, if set it must be one of “Stock” (the default), “Index”, “ETF” or “REIT”. Default is unset via the NA character value. This field may require the premium subscription.
dp	(optional, numeric) The number of decimal places returned on floating point numbers. The value can be between 0 and 11, with a default value of 5.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by tools::R_user_dir (for R 4.0.0 or later), or the TWELVEDATA_API_KEY variable.

**Details**

This is lightweight accessor which also returns the price. See `get_quote()` for a richer return set.

The function has been named `get_price()` to be consistent with the `get_quote()` function.

**Value**

The requested data is returned.

**Author(s)**

Dirk Eddelbuettel

**See Also**

<https://twelvedata.com/docs>

get\_quote

*Quote Data Accessor for 'twelvedata'***Description**

Retrieve Securities Quotes Data from 'twelvedata'

**Usage**

```
get_quote(
  sym,
  interval = c("1min", "5min", "15min", "30min", "45min", "1h", "2h", "4h", "1day",
    "1week", "1month"),
  as = c("data.frame", "raw"),
  exchange = "",
  country = "",
  volume_time_period = 9,
  type = c(NA_character_, "Stock", "Index", "ETF", "REIT"),
  dp = 5,
  timezone = NA_character_,
  apikey
)
```

**Arguments**

sym	(character) A (single or vector) symbol understood by the backend as a stock symbol, foreign exchange pair, or more. See the 'twelvedata' documentation for details on what is covered.
interval	(character) A valid interval designator ranging from "1min" to "1month". Currently supported are 1, 5, 15, 30 and 45 minutes, 1, 2, 4 hours (using suffix 'h', as well as "1day", "1week" and "1month".
as	(optional, character) A selector for the desired output format: one of "data.frame" (the default) or "raw".
exchange	(optional, character) A selection of the exchange for which data for "sym" is requested, default value is unset.
country	(optional, character) A selection of the country exchange for which data for "sym" is requested, default value is unset.
volume_time_period	(optional, numeric) The number of days to use when computing average volume, default is 9.
type	(optional, character) A valid security type selection, if set it must be one of "Stock" (the default), "Index", "ETF" or "REIT". Default is unset via the NA character value. This field may require the premium subscription.
dp	(optional, numeric) The number of decimal places returned on floating point numbers. The value can be between 0 and 11, with a default value of 5.

timezone	(optional, character) The timezone of the returned time stamp. This parameter is optional. Possible values are “Exchange” (the default) to return the exchange-supplied value, “UTC” to use UTC, or a value IANA timezone name such as “America/New_York” (see <code>link{OlsonNames}</code> to see the values R knows). Note that the IANA timezone values are case-sensitive.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by <code>tools::R_user_dir</code> (for R 4.0.0 or later), or the <code>TWELVEDATA_API_KEY</code> variable.

### Details

The function has been named `get_quote()` to not clash with the base R function `quote`.

### Value

The requested data is returned as a `data.frame` object with as many rows as there were symbols in the request.

### Author(s)

Dirk Eddelbuettel

### See Also

<https://twelvedata.com/docs>

---

ref\_crypto\_exchanges    *Reference Data Accessor for Crypto Exchanges from ‘twelvedata’*

---

### Description

Retrieve Reference Data for Crypto Exchanges from ‘twelvedata’

### Usage

```
ref_crypto_exchanges(as = c("data.frame", "raw"), apikey)
```

### Arguments

as	(optional, character) A selector for the desired output format: one of “data.frame” (the default) or or “raw”.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by <code>tools::R_user_dir</code> (for R 4.0.0 or later), or the <code>TWELVEDATA_API_KEY</code> variable.

**Details**

ref\_crypto\_exchanges.

**Value**

The requested data is returned as a `data.frame` object.

**See Also**

<https://twelvedata.com/docs>

---

ref_crypto_pairs	<i>Reference Data Accessor for Forex Pairs from ‘twelvedata’</i>
------------------	------------------------------------------------------------------

---

**Description**

Retrieve Reference Data for Cryptocurrency Pairs from ‘twelvedata’

**Usage**

```
ref_crypto_pairs(
  sym = "",
  as = c("data.frame", "raw"),
  exchange = "",
  currency_base = "",
  currency_quote = "",
  flatten_exchanges = TRUE,
  apikey
)
```

**Arguments**

sym	(optional, character) A (single or vector) Crypto currency pairs with slash(/) delimiter.
as	(optional, character) A selector for the desired output format: one of “data.frame” (the default) or “raw”.
exchange	(optional, character) Exchange where crypto is traded. Default value is unset.
currency_base	(optional, character) Base currency name. Default value is unset.
currency_quote	(optional, character) Quote currency name. Default value is unset.
flatten_exchanges	(bool) Flatten the “data.frame”.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by <code>tools::R_user_dir</code> (for R 4.0.0 or later), or the <code>TWELVEDATA_API_KEY</code> variable.



**Details**

ref\_crypto\_pairs.

**Value**

The requested data is returned as a `data.frame` object.

**See Also**

<https://twelvedata.com/docs>

---

ref\_earliest\_timestamp

*Reference Data Accessor for Symbol Search from 'twelvedata'*

---

**Description**

Retrieve Reference Data for Symbol Search from 'twelvedata'

**Usage**

```
ref_earliest_timestamp(
  sym,
  interval = c("1min", "5min", "15min", "30min", "45min", "1h", "2h", "4h", "1day",
    "1week", "1month"),
  as = c("data.frame", "raw"),
  exchange = "",
  timezone = NA_character_,
  apikey
)
```

**Arguments**

sym	(character) A (single or vector) symbol understood by the backend as a stock symbol, foreign exchange pair, or more. See the 'twelvedata' documentation for details on what is covered. In the case of a vector of arguments a list is returned.
interval	(character) A valid interval designator ranging from "1min" to "1month". Currently supported are 1, 5, 15, 30 and 45 minutes, 1, 2, 4 hours (using suffix 'h', as well as "1day", "1week" and "1month".
as	(optional, character) A selector for the desired output format: one of "data.frame" (the default) or or "raw".
exchange	(optional, character) A selection of the exchange for which data for "sym" is requested, default value is unset.

timezone	(optional, character) The timezone of the returned time stamp. This parameter is optional. Possible values are “Exchange” (the default) to return the exchange-supplied value, “UTC” to use UTC, or a value IANA timezone name such as “America/New_York” (see <code>link{OlsonNames}</code> to see the values R knows). Note that the IANA timezone values are case-sensitive. Note that intra-day data is converted to an R datetime object (the standard <code>POSIXct</code> type) using the exchange timestamp in the returned metadata, if present.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by <code>tools::R_user_dir</code> (for R 4.0.0 or later), or the <code>TWELVEDATA_API_KEY</code> variable.

### Details

`ref_earliest_timestamp`.

### Value

The requested data is returned as a `data.frame` object.

### See Also

<https://twelvedata.com/docs>

---

ref\_etf

*Reference Data Accessor for ETFs from ‘twelvedata’*

---

### Description

Retrieve Reference Data for ETFs from ‘twelvedata’

### Usage

```
ref_etf(sym = "", as = c("data.frame", "raw"), apikey)
```

### Arguments

sym	(optional, character) A (single or vector) symbol for ETFs. Defaults to unset.
as	(optional, character) A selector for the desired output format: one of “data.frame” (the default) or “raw”.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by <code>tools::R_user_dir</code> (for R 4.0.0 or later), or the <code>TWELVEDATA_API_KEY</code> variable.

### Details

`ref_etf`.

**Value**

The requested data is returned as a `data.frame` object.

**See Also**

<https://twelvedata.com/docs>

---

ref_exchanges	<i>Reference Data Accessor for Exchanges from 'twelvedata'</i>
---------------	----------------------------------------------------------------

---

**Description**

Retrieve Reference Data for Exchanges from 'twelvedata'

**Usage**

```
ref_exchanges(
  type = c("stock", "etf", "index"),
  as = c("data.frame", "raw"),
  name = "",
  code = "",
  country = "",
  apikey
)
```

**Arguments**

type	(character) A (single) type for exchange for "stock", "etf" or "index". Defaults to "stock".
as	(optional, character) A selector for the desired output format: one of "data.frame" (the default) or "raw".
name	(optional, character) Filter by exchange. Default is unset.
code	(optional, character) Filter by exchange mic code. default is unset.
country	(optional, character) Filter by country name or alpha code. Default is unset.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by <code>tools::R_user_dir</code> (for R 4.0.0 or later), or the <code>TWELVEDATA_API_KEY</code> variable.

**Details**

ref\_exchanges.

**Value**

The requested data is returned as a `data.frame` object.

**See Also**

<https://twelvedata.com/docs>

---

ref\_forex\_pairs

*Reference Data Accessor for Forex Pairs from ‘twelvedata’*


---

**Description**

Retrieve Reference Data for Forex Pairs from ‘twelvedata’

**Usage**

```
ref_forex_pairs(
  sym = "",
  as = c("data.frame", "raw"),
  currency_base = "",
  currency_quote = "",
  apikey
)
```

**Arguments**

sym	(optional, character) A (single or vector) currency pair according to ISO 4217 standard codes with slash(/) delimiter.
as	(optional, character) A selector for the desired output format: one of “data.frame” (the default) or or “raw”.
currency_base	(optional, character) Base currency name. Default value is unset.
currency_quote	(optional, character) Quote currency name. Default value is unset.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by tools::R_user_dir (for R 4.0.0 or later), or the TWELVEDATA_API_KEY variable.

**Details**

ref\_forex\_pairs.

**Value**

The requested data is returned as a `data.frame` object.

**See Also**

<https://twelvedata.com/docs>

---

ref_indices	<i>Reference Data Accessor for Indices from 'twelvedata'</i>
-------------	--------------------------------------------------------------

---

**Description**

Retrieve Reference Data for Indices from 'twelvedata'

**Usage**

```
ref_indices(sym = "", as = c("data.frame", "raw"), country = "", apikey)
```

**Arguments**

sym	(optional, character) A (single or vector) symbol for Indices. Defaults to unset.
as	(optional, character) A selector for the desired output format: one of "data.frame" (the default) or "raw".
country	(optional, character) An alpha code or country name. Defaults to unset.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by tools::R_user_dir (for R 4.0.0 or later), or the TWELVEDATA_API_KEY variable.

**Details**

ref\_indices.

**Value**

The requested data is returned as a `data.frame` object.

**See Also**

<https://twelvedata.com/docs>

---

ref_stocks	<i>Reference Data Accessor for Stocks from 'twelvedata'</i>
------------	-------------------------------------------------------------

---

**Description**

Retrieve Reference Data for Stocks from 'twelvedata'

**Usage**

```
ref_stocks(
  sym = "",
  as = c("data.frame", "raw"),
  exchange = "",
  country = "",
  type = c(NA_character_, "EQUITY", "Common", "Common Stock",
    "American Depositary Receipt", "Real Estate Investment Trust (REIT)", "Unit", "GDR",
    "Closed-end Fund", "ETF", "Depositary Receipt", "Preferred Stock",
    "Limited Partnership", "OTHER_SECURITY_TYPE", "Warrant", "STRUCTURED_PRODUCT",
    "Exchange-traded Note", "Right", "FUND", "Trust", "Index",
    "Unit Of Beneficial Interest", "MUTUALFUND", "New York Registered Shares"),
  apikey
)
```

**Arguments**

sym	(optional, character) A (single or vector) symbol understood by the backend as a stock symbol, foreign exchange pair, or more. See the ‘twelvedata’ documentation for details on what is covered.
as	(optional, character) A selector for the desired output format: one of “data.frame” (the default) or “raw”.
exchange	(optional, character) A selection of the exchange. Default value is unset.
country	(optional, character) A selection of the country exchanges. Default value is unset.
type	(optional, character) A valid security type selection, if set it must be one of <code>formals(ref_stock)\$type</code> , with NA as default.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by <code>tools::R_user_dir</code> (for R 4.0.0 or later), or the <code>TWELVEDATA_API_KEY</code> variable.

**Details**

`ref_stocks`.

**Value**

The requested data is returned as a `data.frame` object.

**See Also**

<https://twelvedata.com/docs>

---

ref_symbol_search	<i>Reference Data Accessor for Symbol Search from 'twelvedata'</i>
-------------------	--------------------------------------------------------------------

---

**Description**

Retrieve Reference Data for Symbol Search from 'twelvedata'

**Usage**

```
ref_symbol_search(sym, as = c("data.frame", "raw"), output_size = 30, apikey)
```

**Arguments**

sym	(required, character) Symbol to search for.
as	(optional, character) A selector for the desired output format: one of "data.frame" (the default) or "raw".
output_size	(required, character) Defaults to 30. Max is 120.
apikey	(optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by <code>tools::R_user_dir</code> (for R 4.0.0 or later), or the <code>TWELVEDATA_API_KEY</code> variable.

**Details**

ref\_symbol\_search.

**Value**

The requested data is returned as a `data.frame` object.

**See Also**

<https://twelvedata.com/docs>

---

store_key	<i>Store API key</i>
-----------	----------------------

---

**Description**

Store the 'twelvedata' API key

**Usage**

```
store_key(apikey)
```

**Arguments**

apikey                      A character variable with the API key

**Details**

An API key is required to access the functionality offered by the ‘twelvedata’. This function can store a copy in a per-user configuration file. It relies on a the base R function `R_user_dir` which offers a per-user configuration directory for the package. No file or directory permissions are set explicitly or changed. As an alternative the key can also be set in the environment variable `TWELVEDATA_API_KEY` but doing so is the responsibility of the user.

**Value**

TRUE is return invisibly but the function is invoked for the side effect of storing the API key.

**Note**

This function requires R version 4.0.0 or later to utilise the per-user config directory accessor function. For older R versions, please use the alternate approach of storing the API in the environment variable `TWELVEDATA_API_KEY`.

**Author(s)**

Dirk Eddelbuettel

---

time\_series

*Time Series Data Accessor for ‘twelvedata’*


---

**Description**

Retrieve Time Series Data from ‘twelvedata’

**Usage**

```
time_series(
  sym,
  interval = c("1min", "5min", "15min", "30min", "45min", "1h", "2h", "4h", "1day",
    "1week", "1month"),
  as = c("data.frame", "xts", "raw"),
  exchange = "",
  country = "",
  type = c(NA_character_, "Stock", "Index", "ETF", "REIT"),
  outputsize = NA_character_,
  dp = 5,
  order = c("ASC", "DESC"),
  timezone = NA_character_,
  start_date = NA_character_,
```



```

    end_date = NA_character_,
    previous_close = FALSE,
    apikey
  )

```

## Arguments

sym	(character) A (single or vector) symbol understood by the backend as a stock symbol, foreign exchange pair, or more. See the ‘twelvedata’ documentation for details on what is covered. In the case of a vector of arguments a list is returned.
interval	(character) A valid interval designator ranging from “1min” to “1month”. Currently supported are 1, 5, 15, 30 and 45 minutes, 1, 2, 4 hours (using suffix ‘h’, as well as “1day”, “1week” and “1month”.
as	(optional, character) A selector for the desired output format: one of “data.frame” (the default), “xts” (requiring the package to be installed), or “raw”.
exchange	(optional, character) A selection of the exchange for which data for “sym” is requested, default value is unset.
country	(optional, character) A selection of the country exchange for which data for “sym” is requested, default value is unset.
type	(optional, character) A valid security type selection, if set it must be one of “Stock” (the default), “Index”, “ETF” or “REIT”. Default is unset via the NA character value. This field may require the premium subscription.
outputsize	(optional, numeric) The requested number of data points with an internal default value of 30 if unset, and a valid range of 1 to 5000; we use NA as a default argument to signify leaving it unset.
dp	(optional, numeric) The number of decimal places returned on floating point numbers. The value can be between 0 and 11, with a default value of 5.
order	(optional, character) The sort order for the returned time series, must be one of “ASC” (the default) or “DESC”.
timezone	(optional, character) The timezone of the returned time stamp. This parameter is optional. Possible values are “Exchange” (the default) to return the exchange-supplied value, “UTC” to use UTC, or a value IANA timezone name such as “America/New_York” (see <a href="#">link{OlsonNames}</a> to see the values R knows). Note that the IANA timezone values are case-sensitive. Note that intra-day data is converted to an R datetime object (the standard POSIXct type) using the exchange timestamp in the returned metadata, if present.
start_date	(optional, character or date(time) type) The beginning of the time window for which data is requested, can be used with or without end_date. The format must be a standard ISO 8601 format such as “2020-12-31” or “2020-12-31 08:30:00”, or a Date or POSIXt object.
end_date	(optional, character or date(time) type) The end of the time window for which data is requested, can be used with or without start_date. The format must be a standard ISO 8601 format such as “2020-12-31” or “2020-12-31 08:30:00”, or a Date or POSIXt object.

**previous\_close** (optional, boolean) A logical switch to select inclusion of the previous close value, defaults to FALSE.

**apikey** (optional character) An API key override, if missing a value cached from package startup is used. The startup looks for either a file in the per-package config directory provided by `tools::R_user_dir` (for R 4.0.0 or later), or the `TWELVEDATA_API_KEY` variable.

## Details

This function access time series data from ‘twelvedata’. It requires an API key to be registered and to be supplied either in user-config file (TODO: add simple writer) or an environment variable.

All suitable optional parameters of the API are now supported. We excluded return as csv as this downloader encompasses that functionality by returning a parsed object.

## Value

The requested data is returned in the requested format containing columns for data(time), open, high, low, close, and volume. If the request was unsuccessful, an error message is returned. The date or datetime column is returned parsed as either a Date or Datetime where the latter is parsed under the exchange timezone if present. Additional meta data returned from the query is also provided as attributes. If the call contained a vector of symbols, a (named) list of elements is returned.

## Author(s)

Dirk Eddelbuettel

## See Also

<https://twelvedata.com/docs>

## Examples

```
## Not run:  # requires API key
Sys.setenv(`_R_S3_METHOD_REGISTRATION_NOTE_OVERWRITES_`="false") # suppress load noise
data <- time_series("SPY", "5min", outputsize=500, as="xts")
if (requireNamespace("quantmod", quietly=TRUE))
  suppressMessages(library(quantmod)) # suppress some noise
chartSeries(data, name=attr(data, "symbol"), theme="white") # convenient plot for OHLCV
str(data) # compact view of data and meta data

cadusd <- time_series(sym="CAD/USD", interval="1week", outputsize=52.25*20, as="xts")
chart_Series(cadusd, name=attr(data, "symbol"))

gme <- time_series("GME", "1min", start_date="2021-01-25T09:30:00",
  end_date="2021-02-04T16:00:00", as="xts")
chart_Series(gme, name=paste0(attr(gme, "symbol"), "/", attr(gme, "exchange"))))

res <- time_series(c("SPY", "QQQ", "IWM", "EEM"), outputsize=300, as="xts")
op <- par(mfrow=c(2,2))
sapply(res, function(x) quantmod::chart_Series(x, name=attr(x, "symbol")))
par(op)
```

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

# Index

## **\* package**

td-package, [2](#)

fun\_profile, [3](#)

fun\_statistics, [3](#)

get\_price, [4](#)

get\_quote, [6](#)

ref\_crypto\_exchanges, [7](#)

ref\_crypto\_pairs, [8](#)

ref\_earliest\_timestamp, [9](#)

ref\_etf, [10](#)

ref\_exchanges, [11](#)

ref\_forex\_pairs, [12](#)

ref\_indices, [13](#)

ref\_stocks, [13](#)

ref\_symbol\_search, [15](#)

store\_key, [15](#)

td (td-package), [2](#)

td-package, [2](#)

time\_series, [16](#)