Package ‘crypto2’
January 25, 2022

Title  Download Crypto Currency Data from 'CoinMarketCap' without 'API'
Version  1.4.3
Date  2022-01-24
Maintainer  Sebastian Stoeckl <sebastian.stoeckl@uni.li>
URL  https://github.com/sstoeckl/crypto2

Description  Retrieves crypto currency information and historical prices as well as information on the exchanges they are listed on. Historical data contains daily open, high, low and close values for all crypto currencies. All data is scraped from <https://coinmarketcap.com> via their 'web-api'.

License  MIT + file LICENSE
Depends  R (>= 4.0.0)
Imports  dplyr, tibble, tidyr, purrr, progress, stats, lubridate, jsonlite, cli, plyr
Encoding  UTF-8
RoxygenNote  7.1.2
Suggests  spelling
Language  en-US
NeedsCompilation  no

Author  Sebastian Stoeckl [aut, cre] (<https://orcid.org/0000-0002-4196-6093>), Package commissioner and maintainer., Jesse Vent [ctb] (Creator of the crypto package that provided the idea and basis for this package.)

Repository  CRAN
Date/Publication  2022-01-25 18:02:44 UTC

R topics documented:

crypto_history  ................................................................. 2
crypto_info  .................................................................. 4
crypto_list  ...................................................................... 5
exchange_info  ................................................................. 7
exchange_list  ................................................................. 8
crypto_history

Description

Scrape the crypto currency historic market tables from `CoinMarketCap` [https://coinmarketcap.com](https://coinmarketcap.com) and display the results in a dataframe/tibble. This can be used to conduct analysis on the crypto financial markets or to attempt to predict future market movements or trends.

Usage

```r
crypto_history(
  coin_list = NULL,
  convert = "USD",
  limit = NULL,
  start_date = NULL,
  end_date = NULL,
  interval = NULL,
  sleep = NULL,
  finalWait = TRUE
)
```

Arguments

- `coin_list` string if NULL retrieve all currently existing coins (crypto_list()), or provide list of crypto currencies in the crypto_list() format (e.g. current and/or dead coins since 2015)
- `convert` (default: USD) to one or more of available fiat or precious metals prices (fiat_list()). If more than one are selected please separate by comma (e.g. "USD,BTC")
- `limit` integer Return the top n records, default is all tokens
- `start_date` string Start date to retrieve data from, format ‘yyyymmdd’
- `end_date` string End date to retrieve data from, format ‘yyyymmdd’, if not provided, today will be assumed
- `interval` string Interval with which to sample data, default 'daily'. Must be one of "hourly" "daily" "weekly" "monthly" "yearly" "1d" "2d" "3d" "7d" "14d" "15d" "30d" "60d" "90d" "365d"
- `sleep` integer Seconds to sleep for between API requests
- `finalWait` to avoid calling the web-api again with another command before 60s are over (TRUE=default)

Value

Crypto currency historic OHLC market data in a dataframe and additional information via attribute "info":

- `timestamp` Timestamp of entry in database
This is the main function of the crypto package. If you want to retrieve ALL active coins then do not pass an argument to `crypto_history()`, or pass the coin name.

**Examples**

```r
## Not run:

# Retrieving market history for ALL crypto currencies
all_coins <- crypto_history(limit = 2)
one_coin <- crypto_history(limit = 1)

# Retrieving market history since 2020 for ALL crypto currencies
all_coins <- crypto_history(start_date = '20200101', limit=10)

# Retrieve 2015 history for all 2015 crypto currencies
coin_list_2015 <- crypto_list(only_active=TRUE) %>%
  dplyr::filter(first_historical_data"2015-12-31",
                last_historical_data="2015-01-01")
coins_2015 <- crypto_history(coin_list = coin_list_2015,
                             start_date = "20150101", end_date="20151231", limit=20, interval="90d")
```
crypto_info

Retrieves info (urls, logo, description, tags, platform, date_added, notice, status) on CMC for given id or slug

Description
This code uses the web api. It retrieves data for all active, delisted and untracked coins! It does not require an 'API' key.

Usage
crypto_info(
  coin_list = NULL,
  limit = NULL,
  requestLimit = 300,
  finalWait = TRUE
)

Arguments

coin_list  string if NULL retrieve all currently active coins (crypto_list()), or provide list of cryptocurrencies in the crypto_list() format (e.g. current and/or dead coins since 2015)
limit      integer Return the top n records, default is all tokens
requestLimit limiting the length of request URLs when bundling the api calls
finalWait  to avoid calling the web-api again with another command before 60s are over (TRUE=default)

Value
List of (active and historically existing) cryptocurrencies in a tibble:

id           CMC id (unique identifier)
name         Coin name
symbol       Coin symbol (not-unique)
slug         Coin URL slug (unique)
category     Coin category: "token" or "coin"
description  Coin description according to CMC
log           CMC url of CC logo
subreddit     Name of subreddit community
notice        Markdown formatted notices from CMC
date_added    Date CC was added to the CMC database
twitter_username Username of CCs twitter account
Examples

```r
## Not run:
# return info for bitcoin
coin_info <- crypto_info(limit=3)
```

## End(Not run)

crypto_list

Retrieves name, CMC id, symbol, slug, rank, an activity flag as well as activity dates on CMC for all coins

description

This code uses the web api. It retrieves data for all historic and all active coins and does not require an 'API' key.

This code uses the web api. It retrieves data for all available fiat currencies and does not require an API key.

usage

crypto_list(only_active = TRUE, add_untracked = FALSE)

fiat_list(include_metals = FALSE)

arguments

only_active Shall the code only retrieve active coins (TRUE=default) or include inactive coins (FALSE)

add_untracked Shall the code additionally retrieve untracked coins (FALSE=default)

include_metals (default FALSE) Shall the code also retrieve available precious metals?
crypto_list

Value

List of (active and historically existing) cryptocurrencies in a tibble:

- **id**: CMC id (unique identifier)
- **name**: Coin name
- **symbol**: Coin symbol (not-unique)
- **slug**: Coin URL slug (unique)
- **rank**: Current rank on CMC (if still active)
- **is_active**: Flag showing whether coin is active (1), inactive(0) or untracked (-1)
- **first_historical_data**: First time listed on CMC
- **last_historical_data**: Last time listed on CMC, today's date if still listed

List of (active and historically existing) cryptocurrencies in a tibble:

- **id**: CMC id (unique identifier)
- **symbol**: Coin symbol (not-unique)
- **name**: Coin name
- **sign**: Fiat currency sign
- **code**: Precious metals code

Examples

```r
## Not run:
# return all coins
active_list <- crypto_list(only_active=TRUE)
all_but_untracked_list <- crypto_list(only_active=FALSE)
full_list <- crypto_list(only_active=FALSE, add_untracked=TRUE)

# return all coins active in 2015
coin_list_2015 <- active_list %>%
dplyr::filter(first_historical_data<="2015-12-31",
               last_historical_data>="2015-01-01")

## End(Not run)

## Not run:
# return all fiat currencies available on CMC as well as precious metals
fiat_list <- fiat_list(include_metals=TRUE)

## End(Not run)
```
exchange_info

Retrieves info (urls, logo, description, tags, platform, date_added, notice, status) on CMC for given exchange slug

Description

This code uses the web api. It retrieves data for all active, delisted and untracked exchanges! It does not require an 'API' key.

Usage

```r
exchange_info(
  exchange_list = NULL,
  limit = NULL,
  requestLimit = 300,
  finalWait = TRUE
)
```

Arguments

- `exchange_list` string if NULL retrieve all currently active exchanges (exchange_list()), or provide list of exchanges in the exchange_list() format (e.g. current and/or delisted)
- `limit` integer Return the top n records, default is all exchanges
- `requestLimit` limiting the length of request URLs when bundling the api calls
- `finalWait` to avoid calling the web-api again with another command before 60s are over (TRUE=default)

Value

List of (active and historically existing) exchanges in a tibble:

- `id` CMC exchange id (unique identifier)
- `name` Exchange name
- `slug` Exchange URL slug (unique)
- `description` Exchange description according to CMC
- `notice` Exchange notice (markdown formatted) according to CMC
- `logo` CMC url of CC logo
- `type` Type of exchange
- `date_launched` Launch date of this exchange
- `is_hidden` TBD
- `is_redistributable` TBD
- `maker_fee` Exchanges maker fee
taker_fee  Exchanges maker fee
spot_volume_usd  Current volume in USD according to CMC
spot_volume_last_updated  Latest update of spot volume
status  timestamp and other status messages
tags  Tibble of tags and tag categories
url  Tibble of various resource urls. Gives website, blog, fee, twitter.
countries  Tibble of countries the exchange is active in
fiat  Tibble of fiat currencies the exchange trades in

Examples

```r
# Not run:
# return info for the first three exchanges
exchange_info <- exchange_info(limit=3)

# End(Not run)
```

---

**exchange_list**

Retrieves name, CMC id, symbol, slug, rank, an activity flag as well as activity dates on CMC for all coins

**Description**

This code uses the web api. It retrieves data for all historic and all active coins and does not require an 'API' key.

**Usage**

```r
exchange_list(only_active = TRUE, add_untracked = FALSE)
```

**Arguments**

- **only_active** Shall the code only retrieve active coins (TRUE=default) or include inactive coins (FALSE)
- **add_untracked** Shall the code additionally retrieve untracked coins (FALSE=default)

**Value**

List of (active and historically existing) cryptocurrencies in a tibble:

- **id**  CMC exchange id (unique identifier)
- **name**  Exchange name
- **slug**  Exchange URL slug (unique)
### exchange_list

- **is_active**: Flag showing whether exchange is active (1), inactive (0) or untracked (-1)
- **first_historical_data**: First time listed on CMC
- **last_historical_data**: Last time listed on CMC, *today's date* if still listed

#### Examples

```r
## Not run:
# return all coins
ex_active_list <- exchange_list(only_active=TRUE)
ex_all_but_untracked_list <- exchange_list(only_active=FALSE)
ex_full_list <- exchange_list(only_active=FALSE, add_untracked=TRUE)

## End(Not run)
```
Index

crypto_history, 2
crypto_info, 4
crypto_list, 5

exchange_info, 7
exchange_list, 8

fiat_list (crypto_list), 5