Package ‘coinmarketcapr’

October 12, 2022

Type  Package
Title  Get Cryptocurrencies' Market Cap Prices from Coin Market Cap
Version  0.4
Description  Extract and monitor price and market cap of 'Cryptocurrencies' from 'Coin Market Cap' <https://coinmarketcap.com/api/>.
URL  https://github.com/amrrs/coinmarketcapr
License  MIT + file LICENSE
Imports  jsonlite, ggplot2, data.table, curl, cli, crayon
Suggests  testthat, knitr
RoxygenNote  7.1.1
Encoding  UTF-8
BugReports  https://github.com/amrrs/coinmarketcapr/issues
NeedsCompilation  no
Author  AbdulMajedRaja RS [aut, cre],
        Sebastian Gatscha [aut, ctb]
Maintainer  AbdulMajedRaja RS <amrrs.data@gmail.com>
Repository  CRAN
Date/Publication  2022-02-27 12:10:02 UTC

R topics documented:

  coinmarketcapr .............................................................. 2
  get_api_info ................................................................. 2
  get_crypto_listings ....................................................... 3
  get_crypto_map .............................................................. 4
  get_crypto_marketpairs ................................................... 5
  get_crypto_meta ............................................................ 6
  get_crypto_ohlc ............................................................. 7
  get_crypto_quotes ......................................................... 8
get_api_info

get_api_info

get_exchange_map .................................................. 9
get_exchange_meta .................................................. 10
get_global_marketcap ................................................. 11
get_price_conversion ............................................... 12
get_valid_currencies ............................................... 13
plot_top_currencies ............................................... 13
setup ............................................................... 14

Index ..................................................................... 15

coinmarketcapr
coinmarketcapr: Cryptocurrency Market Cap Prices from CoinMarketCap

Description

Extract and monitor price and market cap of ‘Cryptocurrencies’ from ‘CoinMarketCap’ https://coinmarketcap.com/api/ that lists many leading cryptocurrencies along with their price, 24h trade volume, market cap and much more in USD and other currencies.

See Also

Useful links:

- https://coinmarketcap.com/api/
- https://github.com/amrrs/coinmarketcapr
- Report bugs at https://github.com/amrrs/coinmarketcapr/issues

get_api_info
Get API-Key Info

Description

Returns API key details and usage stats. This endpoint can be used to programmatically monitor your key usage compared to the rate limit and daily/monthly credit limits available to your API plan. You may use the Developer Portal’s account dashboard as an alternative to this endpoint.

Usage

get_api_info()

Value

A dataframe with all API key infos
get_crypto_listings

References

API documentation

Examples

```r
## Not run:
get_api_info()

## End(Not run)
```

---

get_crypto_listings  Get latest/historical market data

Description

Get a paginated list of all active cryptocurrencies with latest market data. The default "market_cap" sort returns cryptocurrency in order of CoinMarketCap’s market cap rank (as outlined in our methodology) but you may configure this call to order by another market ranking field. Use the "convert" option to return market values in multiple fiat and cryptocurrency conversions in the same call.

Usage

```r
get_crypto_listings(currency = "USD", latest = TRUE, ...)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>currency</td>
<td>currency code - Default is ’USD’</td>
</tr>
<tr>
<td>latest</td>
<td>If ‘TRUE’ (default), only the latest data is retrieved, otherwise historical data is returned. (NOTE: Historic Data require higher API rights)</td>
</tr>
<tr>
<td>...</td>
<td>Further arguments can be passed to historical data. Further information can be found in the API documentation</td>
</tr>
</tbody>
</table>

Value

A dataframe of top Cryptocurrencies with current or historic market data

References

API documentation

See Also

Other Cryptocurrencies: `get_crypto_map()`, `get_crypto_marketpairs()`, `get_crypto_meta()`, `get_crypto_ohlc()`, `get_crypto_quotes()`, `get_marketcap_ticker_all()`, `get_valid_currencies()`
get_crypto_map

Examples

## Not run:
get_crypto_listings('EUR')
get_crypto_listings('GBP')
get_crypto_listings('GBP', latest=F, start=1,
                 date=Sys.Date()-20, limit=10, sort="price", sort_dir="asc")

## End(Not run)

get_crypto_map

Get active cryptocurrencies

Description

Get all active cryptocurrencies supported by the platform including a unique id

Usage

get_crypto_map(...)  

Arguments

... Further arguments passed to the request. Further information can be found in the API documentation

Value

A dataframe with all active cryptocurrencies supported by the platform including a unique id for each cryptocurrency.

References

API documentation

See Also

Other Cryptocurrencies: get_crypto_listings(), get_crypto_marketpairs(), get_crypto_meta(), get_crypto_ohlc.lines(), get_crypto_quotes(), get_marketcap_ticker_all(), get_valid_currencies()

Examples

## Not run:
get_crypto_map()
get_crypto_map(symbol="BTC")
get_crypto_map(symbol=c("BTC","ETH"))
get_crypto_map(listing_status = "active", start = 1, limit = 10)
get_crypto_map(listing_status = "inactive", start = 1, limit = 10)

## End(Not run)
get_crypto_marketpairs

List all active market pairs

Description

Get a list of all active market pairs that CoinMarketCap tracks for a given cryptocurrency or fiat currency

Usage

get_crypto_marketpairs(
  currency = "USD",
  symbol = NULL,
  id = NULL,
  slug = NULL,
  start = NULL,
  limit = NULL
)

Arguments

currency  currency code - Default is 'USD'
symbol    One or more cryptocurrency symbols. Example: c("BTC","ETH").
id        Alternatively pass one or more CoinMarketCap cryptocurrency IDs. Example: c(1,2)
slug      Alternatively pass a vector of exchange slugs. Example: c("binance","cryptsy")
start     Optionally offset the start (1-based index) of the paginated list of items to return.
          - Default is 1
limit     Optionally specify the number of results to return. Use this parameter and the
          "start" parameter to determine your own pagination size.

Value

A dataframe with all active market pairs

Note

A single cryptocurrency "id", "slug", or "symbol" is required.

References

API documentation
get_crypto_meta

See Also

Other Cryptocurrencies: get_crypto_listings(), get_crypto_map(), get_crypto_meta(), get_crypto_ohlcv(), get_crypto_quotes(), get_marketcap_ticker_all(), get_valid_currencies()

Examples

```r
## Not run:
get_crypto_marketpairs("EUR")
get_crypto_marketpairs("EUR", slug = "bitcoin")
get_crypto_marketpairs("EUR", symbol = "LTC")
get_crypto_marketpairs("EUR", symbol = "BTC", start = 10, limit = 20)
```

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

---

get_crypto_meta  Get static metdata

Description

Get all static metdata available for one or more cryptocurrencies

Usage

```r
get_crypto_meta(symbol = NULL, id = NULL, slug = NULL)
```

Arguments

- **symbol**: One or more cryptocurrency symbols. Example: c("BTC","ETH").
- **id**: Alternatively pass one or more CoinMarketCap cryptocurrency IDs. Example: c(1,2)
- **slug**: Alternatively pass a vector of exchange slugs. Example: c("binance","cryptsy")

Value

A dataframe with metadata of Cryptocurrencies

Note

At least one "id" or "slug" or "symbol" is required for this request.

References

API documentation

See Also

Other Cryptocurrencies: get_crypto_listings(), get_crypto_map(), get_crypto_marketpairs(), get_crypto_ohlcv(), get_crypto_quotes(), get_marketcap_ticker_all(), get_valid_currencies()
**get_crypto_ohlcv**

### Examples

```r
## Not run:
get_crypto_meta()
get_crypto_meta(symbol = c("BTC","ETH"))
get_crypto_meta(id = c(1,2,3,4))
get_crypto_meta(slug = c("bitcoin", "ethereum"))

## End(Not run)
```

---

**get_crypto_ohlcv**  
_List latest/historical OHLCV values_

### Description

Return the latest/historical OHLCV (Open, High, Low, Close, Volume) market values for one or more cryptocurrencies for the current UTC day. Since the current UTC day is still active these values are updated frequently. You can find the final calculated OHLCV values for the last completed UTC day along with all historic days using /cryptocurrency/ohlcv/historical.

### Usage

```r
get_crypto_ohlcv(
  currency = "USD",
  latest = TRUE,
  symbol = NULL,
  id = NULL,
  ...
)
```

### Arguments

- **currency**  
  currency code - Default is ‘USD’

- **latest**  
  If ‘TRUE’ (default), only the latest data is retrieved, otherwise historical data is returned.

- **symbol**  
  One or more cryptocurrency symbols. Example: c("BTC","ETH").

- **id**  
  Alternatively pass one or more CoinMarketCap cryptocurrency IDs. Example: c(1,2)

- **...**  
  Further arguments passed to the request. Further information can be found in the [API documentation](#).

### Value

A dataframe with OHLCV values

### Note

One of "id" or "symbol" is required for this request.
get_crypto_quotes

References

API documentation

See Also

Other Cryptocurrencies: get_crypto_listings(), get_crypto_map(), get_crypto_marketpairs(), get_crypto_meta(), get_crypto_quotes(), get_marketcap_ticker_all(), get_valid_currencies()

Examples

```r
## Not run:
get_crypto_ohlcv("EUR")
get_crypto_ohlcv("EUR", latest = F)
get_crypto_ohlcv("EUR", latest = F, time_period = "hourly",
                  time_start=Sys.Date()-180, count=5, interval="monthly")

## End(Not run)
```

get_crypto_quotes Get market quotes

Description

Get the latest/historical market quotes for 1 or more cryptocurrencies

Usage

```r
get_crypto_quotes(
  currency = "USD",
  symbol = NULL,
  slug = NULL,
  id = NULL,
  latest = TRUE,
  ...
)
```

Arguments

currency        currency code - Default is ‘USD’
symbol          One or more cryptocurrency symbols. Example: c("BTC","ETH").
slug            Alternatively pass a vector of exchange slugs. Example: c("binance","cryptsy")
id               Alternatively pass one or more CoinMarketCap cryptocurrency IDs. Example: c(1,2)
latest          If ‘TRUE’ (default), only the latest data is retrieved, otherwise historical data is returned. (NOTE: Historic Data require higher API rights)
...              Further arguments can be passed to historical data. Further information can be found in the API documentation
get_exchange_map

Value

A dataframe with the latest market quote for 1 or more cryptocurrencies

Note

At least one "id" or "slug" or "symbol" is required for this request.

References

API documentation

See Also

Other Cryptocurrencies: get_crypto_listings(), get_crypto_map(), get_crypto_marketpairs(), get_crypto_meta(), get_crypto_ohlcv(), get_marketcap_ticker_all(), get_valid_currencies()

Examples

## Not run:
get_crypto_quotes()
get_crypto_quotes(symbol="ETH")
get_crypto_quotes(symbol=c("ETH","BTC"))
get_crypto_quotes(slug=c("litecoin","dogecoin"))
get_crypto_quotes("EUR", id=c(3,4))
get_crypto_quotes(latest = FALSE, symbol = c("BTC","ETH"),
  time_start = Sys.Date()-180, time_end=Sys.Date(), count = 10,
  interval = "30m"
)

## End(Not run)

get_exchange_map

Get all cryptocurrency exchanges

Description

Returns a paginated list of all cryptocurrency exchanges by CoinMarketCap ID. We recommend using this convenience endpoint to lookup and utilize our unique exchange id across all endpoints as typical exchange identifiers may change over time. As a convenience you may pass a comma-separated list of exchanges by slug to filter this list to only those you require.

Usage

get_exchange_map(...)

Arguments

... Further arguments passed to the request. Further information can be found in the API documentation
get_exchange_meta

Description
Returns all static metadata for one or more exchanges. This information includes details like launch date, logo, official website URL, social links, and market fee documentation URL.

Usage
get_exchange_meta(id = NULL, slug = NULL)

Arguments
- **id**
  - Alternatively pass one or more CoinMarketCap cryptocurrency IDs. Example: c(1,2)
- **slug**
  - Alternatively pass a vector of exchange slugs. Example: c("binance","cryptsy")

Value
A dataframe with exchange metadata values

References
API documentation

See Also
Other Exchanges: get_exchange_map()
get_global_marketcap

Examples

```r
## Not run:
get_exchange_meta(id = 5)
get_exchange_meta(slug = c("binance", "cryptsy"))

## End(Not run)
```

---

get_global_marketcap  Extract Global Market Cap of Cryptocurrency Market

Description

Extract Global Market Cap of Cryptocurrency Market

Usage

```
get_global_marketcap(currency = "USD", latest = TRUE, ...)
```

Arguments

- `currency`: currency code - Default is 'USD'
- `latest`: If 'TRUE' (default), only the latest data is retrieved, otherwise historical data is returned. (NOTE: Historic Data require higher API rights)
- `...`: Further arguments can be passed to historical data. Further information can be found in the API documentation

Value

A dataframe with global market cap of Cryptocurrencies

Examples

```r
## Not run:
get_global_marketcap('AUD')
get_global_marketcap('EUR')
get_global_marketcap(latest = FALSE, count = 10, interval = "yearly",
                     time_start = Sys.Date()-180, time_end = Sys.Date())

## End(Not run)
```
Description

Convert an amount of one cryptocurrency or fiat currency into one or more different currencies utilizing the latest market rate for each currency. You may optionally pass a historical timestamp as time to convert values based on historical rates (as your API plan supports).

Usage

get_price_conversion(amount = NULL, id, symbol, time, convert, convert_id)

Arguments

amount           An amount of currency to convert. Example: 10.43
id               The CoinMarketCap currency ID of the base cryptocurrency or fiat to convert from. If id and symbol are both missing or NULL, ‘BTC’ is set as default symbol.
symbol           Alternatively the currency symbol of the base cryptocurrency or fiat to convert from. One id or symbol is required.
time             Optional timestamp to reference historical pricing during conversion. If not passed, the current time will be used. If passed, we’ll reference the closest historic values available for this conversion.
convert          Pass up to 120 comma-separated fiat or cryptocurrency symbols to convert the source amount to. Default is ‘USD’.
convert_id       Optionally calculate market quotes by CoinMarketCap ID instead of symbol. This option is identical to convert outside of ID format. Ex: convert_id=1,2781 would replace convert=BTC,USD in your query. This parameter cannot be used when convert is used.

Details

Cache / Update frequency: Every 60 seconds for the lastest cryptocurrency and fiat currency rates. Plan credit use: 1 call credit per call and 1 call credit per convert option beyond the first. CMC equivalent pages: Our cryptocurrency conversion page at converter.

Value

A dataframe with price conversion information

References

API documentation
**get_valid_currencies**

### Examples

```r
## Not run:
get_price_conversion(1)
get_price_conversion(amount = 1, symbol = "BTC", convert = c("EUR","LTC","USD"))
get_price_conversion(amount = 1, id=1, time = Sys.Date()-100)
## End(Not run)
```

---

**get_valid_currencies**  
*Get Valid Currencies*

#### Description

Get Valid Currencies

#### Usage

```r
get_valid_currencies()
```

#### Value

A character vector of valid currencies supported by coinmarketcap API

#### See Also

Other Cryptocurrencies: `get_crypto_listings()`, `get_crypto_map()`, `get_crypto_marketpairs()`, `get_crypto_meta()`, `get_crypto_ohlc()`, `get_crypto_quotes()`, `get_marketcapTicker_all()`

#### Examples

```r
get_valid_currencies()
```

---

**plot_top_currencies**  
*Plot The Price of the Largest Market Cap Cryptocurrencies (API Key required)*

#### Description

Plot The Price of the Largest Market Cap Cryptocurrencies (API Key required)

#### Usage

```r
plot_top_currencies(currency = "USD", k = 5, bar_color = "grey")
```
arguments

- `currency`: currency code (default is 'USD')
- `k`: the number of top cryptocurrencies to plot (default is 5)
- `bar_color`: a valid color name or hexadecimal color code (default is 'grey')

value

A ggplot of top Cryptocurrencies based on their rank (Market Cap)

examples

```r
## Not run:
plot_top_currencies('EUR')
plot_top_currencies('GBP')

## End(Not run)
```

setup

Description

Specifies API Key and the base URL for session

Usage

```r
setup(api_key = NULL, sandbox = FALSE)
get_setup()
reset_setup(api_key = TRUE, sandbox = TRUE)
```

Arguments

- `api_key`: Your Coinmarketcap API key.
- `sandbox`: Sets the base URL for the API. If set to TRUE, the sandbox-API is called. The default is FALSE.

Examples

```r
setup("xXXXXxxxXXXxx")
get_setup()
```
Index

* Cryptocurrencies
  get_crypto_listings, 3
  get_crypto_map, 4
  get_crypto_marketpairs, 5
  get_crypto_meta, 6
  get_crypto_ohlcv, 7
  get_crypto_quotes, 8
  get_valid_currencies, 13

* Exchanges
  get_exchange_map, 9
  get_exchange_meta, 10

* Global Metrics
  get_global_marketcap, 11

* Key
  get_api_info, 2

* Plotting
  plot_top_currencies, 13

* Setup
  setup, 14

* Tools
  get_price_conversion, 12

coinmarketcapr, 2

get_api_info, 2
get_crypto_listings, 3, 4, 6, 8, 9, 13
get_crypto_map, 3, 4, 6, 8, 9, 13
get_crypto_marketpairs, 3, 4, 5, 6, 8, 9, 13
get_crypto_meta, 3, 4, 6, 8, 9, 13
get_crypto_ohlcv, 3, 4, 6, 7, 9, 13
get_crypto_quotes, 3, 4, 6, 8, 13
get_exchange_map, 9, 10
get_exchange_meta, 10, 10
get_global_marketcap, 11
get_marketcap_ticker_all, 3, 4, 6, 8, 9, 13
get_price_conversion, 12
get_setup(setup), 14
get_valid_currencies, 3, 4, 6, 8, 9, 13
plot_top_currencies, 13

reset_setup(setup), 14
setup, 14