Package ‘fmpcloudr’

October 13, 2022

Title R Access to the 'FMP Cloud' and 'Financial Modeling Prep' API
Version 0.1.5
Maintainer Anthony Balentine <exploringfinance1@gmail.com>
URL https://exploringfinance.github.io/fmpcloudr/
BugReports https://github.com/exploringfinance/fmpcloudr/issues
Description Use R to access to the 'FMP Cloud' API <https://fmpcloud.io/> and
Data available includes stock prices, market indexes, company fundamentals,
13F holdings data, and much more. A valid API token must be set to enable
functions.
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Imports jsonlite, httr, dplyr, magrittr, tidyr, lubridate
Suggests testthat
NeedsCompilation no
Author Anthony Balentine [aut, cre]
Repository CRAN
Date/Publication 2021-03-01 18:50:02 UTC

R topics documented:

  fmpe_13f_cik_list .............................................. 2
  fmpe_13f_cik_name ............................................. 3
  fmpe_13f_cik_search .......................................... 4
  fmpe_13f_data ................................................... 4
  fmpe_analyst_outlook .......................................... 5
  fmpe_calendar_events ......................................... 6
  fmpe_cots_data ................................................. 7
List of all 13F ciks

Ciks are unique IDs for each company that files a 13F. This function returns a list of each unique cik that FMP Cloud has available.
**fmpe_13f_cik_name**

**Usage**

```
fmpc_13f_cik_list()
```

**Value**

cik list

**Examples**

```r
## Not run:
# Function can work without a valid API token
fmpc_set_token() # defaults to 'demo'
fmpc_13f_cik_list()

## End(Not run)
```

---

**fmpe_13f_cik_name**  Find name of company by cik

**Description**

Ciks are unique IDs for each company that files a 13F. This function returns the name of the company associated with one or more ciks

**Usage**

```
fmpc_13f_cik_name(cik = "0001067983")
```

**Arguments**

- **cik** a valid cik

**Value**

company name associated with cik

**Examples**

```r
## Not run:
# Function can work without a valid API token
fmpc_set_token() # defaults to 'demo'
fmpc_13f_cik_name(cik = '0001067983')

## End(Not run)
```
**fmpc_13f_cik_search**  
*Search for 13F ciks*

**Description**

Ciks are unique IDs for each company that files a 13F. This function allows search queries to identify and search for specific companies.

**Usage**

```r
fmpc_13f_cik_search(query = "Berkshire")
```

**Arguments**

- `query`  
  a search string to find asset managers by name

**Value**

- cik list

**Examples**

```r
## Not run:
# Function can work without a valid API token
fmpc_set_token() # defaults to 'demo'
fmpc_13f_cik_search('Berkshire')

# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
fmpc_13f_cik_search('Morgan')

## End(Not run)
```

**fmpc_13f_data**  
*13F data for a set of ciks and dates*

**Description**

13F shows the holdings of an asset manager as of a specific date

**Usage**

```r
fmpc_13f_data(ck = "0001067983", date = "2019-12-31")
```
Arguments

cik   one or more valid ciks from fmpc_13f_cik_list

date  one or more quarter end dates (yyyy-03-31, yyyy-06-30, yyyy-09-30, yyyy-12-31)

Value

13F holdings

Examples

## Not run:

```r
# Berkshire can be run using the demo
fmpc_set_token()
fmpc_13f_data('0001067983', '2020-03-31')

# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
goldmanCik = fmpc_13f_cik_search('golman')
fmpc_13f_data(cik = goldmanCik$cik, date = c('2020-03-31', '2020-06-30'))
```

## End(Not run)

---

fmpc_analyst_outlook  Pull analyst outlook for one or more securities

Description

Analyst details for one or more securities includes earnings estimates, earnings surprises, stock grade, analyst recommendations, and company press releases

Usage

```r
fmpc_analyst_outlook(
  symbols = c("AAPL"),
  outlook = c("surprise", "grade", "estimate", "recommend", "press"),
  limit = 100
)
```
fmpc_calendar_events

Arguments

symbols    one or more symbols from the FMP available list that can be found using fmpc_symbols_available. A valid API token must be set in order to enable functionality. See documentation for setting a token under fmpc_set_token.

outlook    can be one of: 'surprise' for earnings surprise, 'grade' for stock grade, 'estimateAnnl' for analysts annual estimates, 'estimateQtr' for analysts quarterly estimate, 'recommend' for analyst recommendations, and 'press' for company press releases

limit      limit results for each ticker

Value

data frame of ratings data

Examples

```r
## Not run:

# Demo can pull AAPL
fmpc_set_token()
fmpc_analyst_outlook('AAPL')

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_analyst_outlook(c('AAPL','MSFT','SPY'))

## End(Not run)
```

fmpc_calendar_events  Dates related to Economic data, IPOs, dividends, earnings, and splits

Description

Calendar dates related to Economic data, IPOs, dividends, earnings, and splits

Usage

```r
fmpc_calendar_events(
    calendar = "economic",
    startDate = Sys.Date() - 180,
    endDate = Sys.Date()
)
```
fmpc_cots_data

Arguments

- calendar: indicate which calendar to view. Options include 'ipo', 'stock_split', 'stock_dividend', 'earning', or 'economic'.
- startDate: filter start date in yyyy-mm-dd format.
- endDate: filter end date in yyyy-mm-dd format.

Value

calendar events

Examples

## Not run:

```r
# Must set a valid API token
fmpc_set_token(Var FMPAPIKEY)
fmpc_calendar_events(Var stock_split)
fmpc_calendar_events(Var economic)
## End(Not run)
```

fmpc_cots_data

Description

Provides raw data from Commitment of Traders report

Usage

```r
fmpc_cots_data(
  cots_sym = "all",
  startDate = Sys.Date() - 30,
  endDate = Sys.Date()
)
```

Arguments

- cots_sym: COTS symbol that can be found in fmpc_cots_symbols. Default is 'all'. Date parameters are only valid with 'all'. Otherwise the full history is pulled.
- startDate: filter start date in yyyy-mm-dd format.
- endDate: filter end date in yyyy-mm-dd format.

Value

a data frame of cots data
Examples

```r
## Not run:

# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
fmpc_cots_data()

## End(Not run)
```

---

### fmpc_cots_symbols

Get list of COTS symbols

**Description**

Provides a list of all the symbols available on the Commitment of Traders data pull

**Usage**

```r
fmpc_cots_symbols()
```

**Value**

a data frame of cots symbols with descriptions

**Examples**

```r
## Not run:

# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
fmpc_cots_symbols()

## End(Not run)
```
**fmpc_cusip_search**

*Use a cusip to search a company and ticker*

**Description**

Use a cusip to search a company and ticker

**Usage**

```r
fmpc_cusip_search(cusip = "000360206")
```

**Arguments**

- `cusip` 
  a valid cusip

**Value**

a data frame of symbol, cusip, and company

**Examples**

```r
## Not run:
# Demo offers AAON as an example
fmpc_set_token()
fmpc_cusip_search("000360206")
## End(Not run)
```

---

**fmpc_earning_call_transcript**

*Full transcript of earnings call*

**Description**

Full transcript of earnings call

**Usage**

```r
fmpc_earning_call_transcript(symbols = c("AAPL"), quarter = 2, year = 2020)
```
Arguments

symbols  one or more symbols from the FMP available list that can be found using `fmpc_symbols_available`. A valid API token must be set in order to enable functionality. See documentation for setting a token under `fmpc_set_token`.

quarter  earnings year quarter

year  earnings call year

Value

earnings call transcript in data frame

Examples

```r
## Not run:

# Demo can pull AAPL
fmpc_set_token()
fmpc_earning_call_transcript('AAPL', quarter = 1, year = 2019)

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_earning_call_transcript(c('AAPL', 'MSFT', 'SPY'))

## End(Not run)
```

---

**fmpc_economic_events**  List economic events that can be searched

Description

Shows the economic events that can be fed into `fmpc_economic_results`

Usage

`fmpc_economic_events()`

Value

economic metrics
Examples

## Not run:
# Function can work without a valid API token
fmpc_set_token('FMPAPIKEY')
fmpc_economic_events()

## End(Not run)

---

fmpc_economic_results Economic results

---

**Description**

Shows the results of the economic events from `fmpc_economic_events`

**Usage**

```r
fmpc_economic_results(
  event = "adpEmploymentChange",
  country = "US",
  startDate = Sys.Date() - 180,
  endDate = Sys.Date()
)
```

**Arguments**

- **event**: an economic event from `fmpc_economic_events`
- **country**: a country from `fmpc_economic_events`
- **startDate**: filter start date in yyyymm-dd format.
- **endDate**: filter end date in yyyymm-dd format.

**Value**

calendar events

**Examples**

## Not run:

# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
Events = fmpc_economic_events()
fmpc_economic_results()

# Italy three month interbank rate
fmpc_economic_results(event = Events[100,1], country = Events[100,2])
## Description

Pull balance sheet, income statement, or cash flow statement for one or more tickers. Can elect to see data quarterly or annually and view growth.

## Usage

```r
fmpc_financial_bs_is_cf(
  symbols = "AAPL",
  statement = c("income", "balance", "cashflow"),
  quarterly = TRUE,
  growth = FALSE,
  SECReported = FALSE,
  limit = 100
)
```

## Arguments

- **symbols**: one or more publicly traded companies domestic or international
- **statement**: indicate the statement to view. Can be 'income', 'balance', or 'cashflow'
- **quarterly**: TRUE/FALSE whether the statement should be shown quarterly or annually
- **growth**: TRUE/FALSE whether the growth statement should be shown
- **SECReported**: TRUE/FALSE whether to display the information as it was reported to the SEC. Only one symbol can be passed if this is set to TRUE. Growth must also be set to FALSE. International stocks should not be passed.
- **limit**: limit the result for each ticker

## Value

returns balance sheet, income statement, or cash flow
## Examples

```r
## Not run:

# Setting API key to 'demo' allows for AAPL only
fmpc_set_token()

fmpc_financial_bs_is_cf()
fmpc_financial_bs_is_cf("AAPL", statement = "balance", quarterly = FALSE, growth = FALSE, SECReported = TRUE, limit = 10)

# Must set a valid API token
fmpc_set_token("FMPAPIKEY")
fmpc_financial_bs_is_cf(c("AAPL", "MSFT", "TSLA"))

symbols = c("AAPL", "MSFT", "BAC")
Bal = fmpc_financial_bs_is_cf(symbols, statement = "balance")
BalG = fmpc_financial_bs_is_cf(symbols, statement = "balance", growth = FALSE)
IS = fmpc_financial_bs_is_cf(symbols, statement = "income")
ISa = fmpc_financial_bs_is_cf(symbols, statement = "income", quarterly = FALSE)
cf = fmpc_financial_bs_is_cf(symbols, statement = "cashflow")
cfsec = fmpc_financial_bs_is_cf(symbols, statement = "cashflow", SECReported = TRUE)

# International tickers work
fmpc_financial_bs_is_cf("RY.TO", statement = "balance", quarterly = TRUE, growth = TRUE, SECReported = FALSE, limit = 10)

## End(Not run)
```

---

### fmpc_financial_dcfv

**Discounted Cash Flow Value**

**Description**

Pull Discounted Cash Flow Value for one or more securities

**Usage**

```r
fmpc_financial_dcfv(
  symbols = c("AAPL"),
  period = c("current", "daily", "quarterly", "annually"),
  limit = 100
)
```
Arguments

symbols  one or more publicly traded companies domestic or international
period  period for discounted cash flow - current, daily, quarterly, annually
limit  limit the result for each ticker

Value

data frame of discounted cash flow

Examples

## Not run:

# Demo can pull AAPL
fmpc_set_token()
fmpc_financial_dcfv('AAPL')

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_financial_dcfv(c('AAPL','MSFT','SPY'), period = 'quarterly')

## End(Not run)

fmpc_financial_metrics

Get financial metrics for one or more tickers

Description

Pull finance ratios, enterprise value, key metrics, and financial growth

Usage

fmpc_financial_metrics(
  symbols = "AAPL",
  metric = c("ratios", "key", "ev", "growth"),
  quarterly = TRUE,
  trailingTwelve = FALSE,
  limit = 100
)
Argsuemts

symbols one or more publicly traded companies domestic or international
metric indicate the statement to view. Can be 'income', 'balance', or 'cashflow'
quartetely TRUE/FALSE whether the statement should be shown quarterly or annually
trailingTwelve TRUE/FALSE whether finance ratios should show as trailing 12 months. Only available for finance ratios.
limit limit the result for each ticker

Value

data frame of financial metrics

Examples

## Not run:

# Setting API key to 'demo' allows for AAPL only
fmpc_set_token()
fmpc_financial_metrics()
fmpc_financial_metrics('AAPL', metric = 'ratios', quarterly = FALSE, trailingTwelve = FALSE, limit = 10)

# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
fmpc_financial_metrics(c('AAPL', 'MSFT', 'TSLA'))

# International tickers work
fmpc_financial_metrics('RY.TO', metric = 'key', quarterly = TRUE, trailingTwelve = TRUE, limit = 10)

## End(Not run)

fmpc_financial_zip

*Get a list of Financial ZIP file locations*

Description

Get location to download ZIP file for all symbols. This function will not download the file. Use a 'wget' command or enter the link into a browser.

Usage

fmpc_financial_zip()
### fmfc_get_url

**Value**

a data frame of symbols with zip file location

**Examples**

```r
## Not run:
# Must set a valid API token
fmpc_set_token("FMPAPIKEY")
fmpc_financial_zip()

## End(Not run)
```

---

**fmfc_get_url**

*GET Request for specific URL*

**Description**

FMP helper function that takes a URL, appends the API Token, makes a GET call, and parses the data

**Usage**

```r
fmpc_get_url(URL, api_version = "3")
```

**Arguments**

- **URL**
  The URL to pull specific data from FMP Cloud. Search parameters should be included, but not the API key. Start with the URL after `api/v3`

- **api_version**
  The API version of the URL. FMP is constantly updating their API and new URLs may be under a new version

**Details**

This function is a helper in most other fmpcloudr functions, so does not need to be used by the end user. That being said, FMP is always adding data with new URLs. If this package does not have a URL available on FMP, this function can be used to simplify the GET call. Pass the URL appearing after `api/v3` with the search parameters entered, but do not include the API token

**Value**

list output of data set
Examples

```r
## Not run:
# Set the FMP Token. The DEMO token has VERY limited access.
fmpc_set_token('demo')

# Pull price history for Apple
AppleHist = fmpc_get_url('historical-price-full/AAPL?serietype=line&')

## End(Not run)
```

---

**fmpc_held_by_mfs**  
List of mutual funds that hold a specified symbol

### Description

Shows mutual funds holding a list of symbols provided. Not currently available for ETFs

### Usage

```r
fmpc_held_by_mfs(symbols = c("AAPL"))
```

### Arguments

- **symbols**: one or more symbols from the FMP available list that can be found using `fmpc_symbols_by_market`. A valid API token must be set in order to enable functionality. See documentation for setting a token under `fmpc_set_token`.

### Details

This differs from `fmpc_holdings_etf` which allows a search by ETF to see detail on what the ETF is holding.

### Value

- data frame of mutual funds holding specified security

### Examples

```r
## Not run:

# Demo can pull AAPL
fmpc_set_token()
fmpc_held_by_mfs('AAPL')

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
```
fmpc_holdings_etf

Pull ETF holdings data

Description

Shows holdings data for one or more ETFs provided. Either individual holdings, sector weightings, or country weightings.

Usage

fmpc_holdings_etf(etfs = c("SPY"), holding = c("symbol", "sector", "country"))

Arguments

etfs

one or more valid ETFs.

holding

holding type can be other individual symbols, the sector breakdown, or the country breakdown. Valid inputs include 'symbol', 'sector', 'country'

Details

This differs from fmpc_held_by_mfs which allows a search by symbol to see the mutual funds holding the specified symbol(s). This function searches by ETF and shows the holding of the ETF

Value

data frame of mutual fund holdings

Examples

## Not run:

# For multiple symbols, set a valid API Token
fmpc_set_token("FMPAPIKEY")
fmpc_holdings_etf(c("VOO","SPY"), holding = 'symbol')
fmpc_holdings_etf(c("VOO","SPY"), holding = 'country')
fmpc_holdings_etf(c("VOO","SPY"), holding = 'sector')

## End(Not run)
fmpc_market_hours  

Current day market hours and holidays

Description
Pulls back current day market hours and holidays for 2019 going forward

Usage
fmpc_market_hours()

Value
an object of class list containing two data frames for market hours and holidays

Examples

## Not run:

# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
fmpc_market_hours()

## End(Not run)

fmpc_price_batch_eod  

Pull all EOD pricing data for a specific date

Description
Returns 10,000 or more symbols for a specific date. This generates one API call for each date. Dates must be entered one at a time.

Usage
fmpc_price_batch_eod(priceDate = Sys.Date() - 1)

Arguments

priceDate  price date to pull all securities. Use format yyyy-mm-dd

Value
data frame of a single days prices
Examples

## Not run:

```r
fmpc_set_token('FMPAPIKEY')
allSymbs = fmpc_price_batch_eod('2020-06-24')
```

## End(Not run)

---

### fmfc_price_current

**Current price**

**Description**

Last traded price with additional price metrics such as 50D and 200D avg price, volume, shares outstanding, and more. This does NOT include a quote, such as bid/ask detail.

**Usage**

```r
fmpc_price_current(symbols = c("AAPL"))
```

**Arguments**

- `symbols` one or more symbols from the FMP available list that can be found using `fmpc_symbols_by_market`. A valid API token must be set in order to enable functionality. See documentation for setting a token under `fmpc_set_token`.

**Value**

- data frame of current price details

**Examples**

## Not run:

```r
# Setting API key to 'demo' allows for AAPL only
fmpc_set_token()
fmpc_price_current('AAPL')
```

```r
# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_price_current(c('AAPL', 'MSFT', 'TSLA', 'SPY', 'BTCUSD', 'JPYUSD', '^SP500TR'))
```

## End(Not run)
fmpc_price_forex

Get all foreign exchange quotes

Description

Use a valid API Token to pull all foreign exchange quotes which includes Bid/Ask. Set API token using `fmpc_set_token`.

Usage

fmpc_price_forex()

Value

a data frame of forex with quotes

Examples

```r
## Not run:
# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
fmpc_price_forex()

## End(Not run)
```

fmpc_price_full_market

Current Price for an Entire Market

Description

Enter a market to get last traded price with additional price metrics such as 50D and 200D avg price, volume, shares outstanding, etc. This does NOT include a quote, such as bid/ask detail. Each market will return multiple responses, but only one API call as made per market.

Usage

fmpc_price_full_market(market = "etf")

Arguments

market Select a market to get current price for all securities: 'etf', 'commodity', 'euronext', 'nyse', 'amex', 'tsx', 'index', 'nasdaq', 'crypto', 'forex'. Select a single market.
fmpc_price_history

Value

data frame of current price

Examples

## Not run:

```r
fmpc_set_token('FMPAPIKEY')
fmpc_price_full_market('index')
fmpc_price_full_market('forex')
fmpc_price_full_market() # Default is to 'etf'
```

## End(Not run)

---

**fmpc_price_history**  
*Historical End of Day pricing data for one or more symbols*

Description

End of Day price history includes open, high, low, close, adjClose, volume, vwap, and more. Symbol can include equity, mutual fund, index, currency, crypto, or any other symbol that can be found in `fmpc_symbols_by_market`.

Usage

```r
fmpc_price_history(
  symbols = "AAPL",
  startDate = Sys.Date() - 30,
  endDate = Sys.Date()
)
```

Arguments

- **symbols**: one or more symbols from the FMP available list that can be found using `fmpc_symbols_by_market`. A valid API token must be set in order to enable functionality. See documentation for setting a token under `fmpc_set_token`.
- **startDate**: filter start date in yyyy-mm-dd format.
- **endDate**: filter end date in yyyy-mm-dd format.

Value

a data frame of price history

Warning

Each symbol is a separate API call. If the API token has monthly limits, this should be considered before making a bulk request.
Examples

```r
## Not run:

# Default sets token to 'demo' which allows for AAPL only
fmpc_set_token()
fmpc_price_history('AAPL')

# For multiple symbols, set a valid API Token
# Crypto, equity, currency, and index can all be entered in the same request
fmpc_set_token('FMPAPIKEY')
fmpc_price_history(c('AAPL','MSFT','SPY','^SP500TR','JPYUSD','BTCUSD'))

## End(Not run)
```

---

### fmpc_price_history_spldiv

**Historical EOD pricing, split, and dividend data for one or more symbols**

---

### Description

Pulls End of Day prices, splits, and dividends for a list of symbols. Symbol can include equity, mutual fund, index, currency, crypto, or any other symbol that can be found in `fmpc_symbols_by_market`.

### Usage

```r
fmpc_price_history_spldiv(
  symbols = "AAPL",
  startDate = Sys.Date() - 360,
  endDate = Sys.Date()
)
```

### Arguments

- **symbols**
  - one or more symbols from the FMP available list that can be found using `fmpc_symbols_by_market`. A valid API token must be set in order to enable functionality. See documentation for setting a token under `fmpc_set_token`.

- **startDate**
  - filter start date in yyyy-mm-dd format.

- **endDate**
  - filter end date in yyyy-mm-dd format.

### Details

Uses functions `fmpc_price_history`, `fmpc_security_dividends`, and `fmpc_security_splits` to aggregate split, dividend, and pricing data into a single data frame. Warnings may be generated that split data or dividend data was not returned. Results will still show in this function if price_history data is available.
**Value**

a data frame of pricing, split, and dividend history

**Warning**

Each symbol is THREE separate API calls. If the API token has monthly limits, this should be considered before making a bulk request. Large Symbol requests will also take time because of the buffer time between API calls.

**Examples**

```r
## Not run:
# Setting API key to 'demo' allows for AAPL only
fmpc_set_token()
fmpc_price_history_spdiv('AAPL')

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
# Index, currency, and crypto will return data even without splits/divs
fmpc_price_history_spdiv(c('AAPL','MSFT','SPY','^SP500TR','JPYUSD','BTCUSD'))

## End(Not run)
```

**fmpc_price_intraday**

*Historical End of Day pricing data for one or more symbols*

**Description**

Intraday includes open, high, low, close, for each time segment. The amount of history available is based on the freq set. Smaller time intervals frequency will pull back less history. Symbol can include equity, mutual fund, index, currency, crypto, or any other symbol that can be found in `fmpc_symbols_by_market`. Available history is limited by the increment size.

**Usage**

```r
fmpc_price_intraday(
  symbols = "AAPL",
  startDate = Sys.Date() - 30,
  endDate = Sys.Date(),
  freq = c("1min", "5min", "15min", "30min", "1hour")
)
```
Arguments

symbols: one or more symbols from the FMP available list that can be found using `fmpc_symbols_by_market`. A valid API token must be set in order to enable functionality. See documentation for setting a token under `fmpc_set_token`.

startDate: filter start date in yyyy-mm-dd format.

d endDate: filter end date in yyyy-mm-dd format.

freq: the frequency of which to pull intraday prices. Options can be '1min', '5min', '15min', '30min', '1hour'

Value

a data frame of intraday prices

Warning

Each symbol is a separate API call. If the API token has monthly limits, this should be considered before making a bulk request

Examples

```r
## Not run:

# Setting API key to 'demo' allows for AAPL only
fmpc_set_token()
# Freq of 1hour will return about 2 months of data
fmpc_price_intraday('AAPL', freq = '5min')

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_price_intraday(symbols = c('AAPL', 'MSFT', 'SPY', '^SP500TR', 'JPYUSD', 'BTCUSD'),
   startDate = '2020-01-01', freq = '1hour')

## End(Not run)
```

fmpc_rss_sec

RSS feed of latest submissions to SEC

Description

RSS feed of latest submissions to SEC that includes 6-k, 10-Q, 13F

Usage

`fmpc_rss_sec(limit = 100)`
Arguments

limit: limit output to a specific number of results

Value

a data frame of title, data, link to submission, cik, and submission type

Examples

```r
## Not run:
# Demo offers AAON as an example
fmpc_set_token()
fmpc_rss_sec()

## End(Not run)
```

---

**fmpc_security_delisted**

*Delisted companies*

Description

Shows companies that have been delisted

Usage

```r
fmpc_security_delisted(limit = 100)
```

Arguments

limit: set limit of results returned

Value

data frame of delisted companies

Examples

```r
## Not run:
# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
fmpc_security_delisted()

## End(Not run)
```
fmpc_security_dividends

Historical security dividend data

Description

Pulls dividends for a list of securities. Data includes dividend, adjusted dividend, payment date, record date, and declaration date.

Usage

```r
fmpc_security_dividends(
  symbols = "AAPL",
  startDate = Sys.Date() - 360,
  endDate = Sys.Date()
)
```

Arguments

- **symbols**: one or more symbols from the FMP available list that can be found using `fmpc_symbols_by_market`. A valid API token must be set in order to enable functionality. See documentation for setting a token under `fmpc_set_token`.
- **startDate**: filter start date in `yyyy-mm-dd` format.
- **endDate**: filter end date in `yyyy-mm-dd` format.

Value

A data frame of dividend history that includes payment date, record date, and declaration date.

Warning

Each symbol is a separate API call. If the API token has monthly limits, this should be considered before making a bulk request.

Examples

```r
## Not run:

# Setting API key to DEMO allows for AAPL only
fmpc_set_token()
fmpc_security_dividends('AAPL')

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_security_dividends(c('AAPL','MSFT','SPY'), startDate = '2010-01-01')
```
fmpc_security_gla  Gainers, Losers, and active

Description
Shows top gainers, bottom losers, and most active for the current trading day

Usage
fmpc_security_gla(gla = c("gainers", "losers", "actives"))

Arguments
  gla  options include 'gainers','losers', and 'active'

Value
securities with details for current trading day

Examples

## Not run:

# Must set a valid API token
fmpc_set_token("FMPAPIKEY")
fmpc_security_gla('gainers')
fmpc_security_gla('losers')
fmpc_security_gla('actives')

## End(Not run)

fmpc_security_mrktcap  Pull market capitalization

Description
Provides current and historical market cap for one or more securities

Usage
fmpc_security_mrktcap(symbols = c("AAPL"), limit = 100)
fmpc_security_news

Arguments

symbols one or more symbols from the FMP available list that can be found using fmpc_symbols_available. A valid API token must be set in order to enable functionality. See documentation for setting a token under fmpc_set_token.

limit limit results for each ticker

Value
data frame of news for tickers

Examples

## Not run:

# Demo can pull AAPL
fmpc_set_token()
fmpc_security_mrktcap('AAPL')

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_security_mrktcap(c('AAPL', 'MSFT', 'SPY'))

## End(Not run)

fmpc_security_news

Get current news

Description

Provides current and historical News. Can enter securities to pull news for specific securities

Usage

fmpc_security_news(symbols = NULL, limit = 100)

Arguments

symbols one or more symbols. Use NULL for general latest news

limit limit results for each ticker

Value
data frame of news for tickers
Examples

```r
## Not run:
# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_security_news(c('AAPL','MSFT','SPY'))
fmpc_security_news()

## End(Not run)
```

---

### fmpc_security_profile

Pull profile for a company

#### Description

Provides pricing, market cap, dividend, company description, CEO, number of employees, and more for one or more symbols

#### Usage

```r
fmpc_security_profile(symbols = c("AAPL"))
```

#### Arguments

- **symbols**: one or more symbols from the FMP available list that can be found using `fmpc_symbols_available`. A valid API token must be set in order to enable functionality. See documentation for setting a token under `fmpc_set_token`.

#### Value

data frame of descriptive information

#### Examples

```r
## Not run:
# Demo can pull AAPL
fmpc_set_token()
fmpc_security_profile('AAPL')

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_security_profile(c('AAPL','MSFT','SPY'))
```
fmpc_security_ratings

## End(Not run)

---

fmpc_security_ratings  *Pull security rating*

### Description

Provides current and historical rating for one or more securities

### Usage

```r
fmpc_security_ratings(symbols = c("AAPL"), limit = 100)
```

### Arguments

- **symbols**
  one or more symbols from the FMP available list that can be found using `fmpc_symbols_available`. A valid API token must be set in order to enable functionality. See documentation for setting a token under `fmpc_set_token`.

- **limit**
  limit results for each ticker

### Value

data frame of ratings data

### Examples

```r
## Not run:
# Demo can pull AAPL
fmpc_set_token()
fmpc_security_ratings('AAPL')

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_security_ratings(c('AAPL','MSFT','SPY'))

## End(Not run)
```
Function: fmpc_security_screener

Description
Filter for stocks based on numerous criteria.

Usage
fmpc_security_screener(
    limit = 100,
    mrktCapAbove = NULL,
    mrktCapBelow = NULL,
    betaAbove = NULL,
    betaBelow = NULL,
    dividendAbove = NULL,
    dividendBelow = NULL,
    volumeAbove = NULL,
    volumeBelow = NULL,
    sector = NULL,
    industry = NULL
)

Arguments
- limit: limit output to a certain amount or rows
- mrktCapAbove: integer - market cap greater than
- mrktCapBelow: integer - market cap less than
- betaAbove: double - beta greater than
- betaBelow: double - beta less than
- dividendAbove: double - dividend greater than
- dividendBelow: double - dividend less than
- volumeAbove: integer - volume in shares greater than
- volumeBelow: integer - volume in shares less than
- sector: indicate a sector. Documentation is unclear of options. 'tech' is the example used
- industry: indicate an industry. Documentation is unclear of options. 'Software' is the example used

Details
This function will temporarily modify options(scipen=999).
Value

a list of securities based on criteria supplied

Examples

## Not run:

# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
fmpc_security_screener() # Default pulls a list of 100 with no filters
# Search for market cap above a billion, 
# that trades at least a million shares with a dividend under 1
fmpc_security_screener(mrktCapAbove = 1e9, dividendBelow = 1, volumeAbove = 1e6)

## End(Not run)

---

**fmpc_security_sector**  
*Pull daily returns for all sectors*

Description

For each day, the return for a sector will be pulled

Usage

```r
fmpc_security_sector(days = 25)
```

Arguments

days  
number of trading days to pull returns from current day. For example, 20 will pull the last 20 trading days, about one month of data.

Value

data frame of sectors and daily returns

Examples

## Not run:

fmpc_set_token('FMPAPIKEY')
allSymbs = fmpc_security_sector(30)

## End(Not run)
fmpc_security_splits  

**Historical security split data**

**Description**
Pulls stock split history for one or more securities

**Usage**
```r
fmpc_security_splits(
  symbols = "AAPL",
  startDate = Sys.Date() - 3600,
  endDate = Sys.Date()
)
```

**Arguments**
- **symbols** one or more symbols from the FMP available list that can be found using `fmpc_symbols_by_market`.
  A valid API token must be set in order to enable functionality. See documentation for setting a token under `fmpc_set_token`.
- **startDate** filter start date in yyyy-mm-dd format.
- **endDate** filter end date in yyyy-mm-dd format.

**Value**
a data frame of split history

**Warning**
Each symbol is a separate API call. If the API token has monthly limits, this should be considered before making a bulk request

**Examples**

```r
## Not run:
# Setting API key to DEMO allows for AAPL only
fmpc_set_token()
fmpc_security_splits('AAPL')

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_security_splits(c('AAPL','TSLA'), startDate = '2020-01-01')

## End(Not run)
```
fmpc_security_tech_indic

Technical Indicators

Description

Pull Technical Indicators for a set of symbols over a specific number of periods set by the frequency

Usage

```r
fmpc_security_tech_indic(
  symbols = "AAPL",
  indicator = c("SMA", "EMA", "WMA", "DEMA", "TEMA", "williams", "RSI", "ADX",
  "standardDeviation"),
  freq = c("daily", "1min", "5min", "15min", "30min", "1hour", "4hour"),
  period = 10
)
```

Arguments

- **symbols**: one or more symbols from the FMP available list that can be found using `fmpc_symbols_by_market`. A valid API token must be set in order to enable functionality. See documentation for setting a token under `fmpc_set_token`.
- **indicator**: 'SMA', 'EMA', 'WMA', 'DEMA', 'TEMA', 'williams', 'RSI', 'ADX', 'standardDeviation'
- **freq**: the frequency of which to pull intraday data. Options are '1min', '5min', '15min', '30min', '1hour'
- **period**: Number of periods over which to calculate the technical indicator

Details


Value

technical indicator in a data frame

Examples

```r
## Not run:
# Setting API key to DEMO allows for AAPL only
fmpc_set_token()
fmpc_security_tech_indic('AAPL')
fmpc_security_tech_indic('AAPL', indicator = 'standardDeviation')
```
fmpc_set_token

fmpc_security_tech_indic('AAPL', 'sma')

# For multiple symbols, set a valid API Token
fmpc_set_token('FMPAPIKEY')
fmpc_security_tech_indic(c('AAPL', 'MSFT', 'SPY'), indicator = 'RSI', freq = '15min', period = 25)

## End(Not run)

---

**fmpc_set_token**  
*Set FMP Token for all other functions*

**Description**

Pass a valid FMP API Token to enable all other functions

**Usage**

fmpc_set_token(APIToken = "demo", timeBtwnReq = 0.1, noBulkWarn = FALSE)

**Arguments**

- **APIToken**: a valid API Token from FMP Cloud
- **timeBtwnReq**: Time between API calls in seconds. FMP Cloud may limit API call frequency depending on the package subscription. Use this to set a time between API calls. The default is .1 to limit calls to no more than 10 per second.
- **noBulkWarn**: Boolean of TRUE or FALSE. When making large API calls, an initial warning will be given to make sure the user wants to make bulk requests. If the user has a FREE FMP API a bulk request may be undesirable. Each symbol requested for many functions will be a unique API request. Use this parameter to silence this warning.

**Details**

In order to use the FMP Cloud API, an account must be created at **FMP Cloud**. There is a free account option that but sets a cap at 250 calls. There are different subscription levels. To use this package, the free subscription can be used up to the limit.

Please note that most functions will make an individual API call for each symbol passed. If running a function on multiple symbols, be aware this could quickly hit API call limits. When running in an interactive environment, a warning will appear asking to confirm running bulk requests. This can be suppressed for a session using the options. You can also suppress when using the option noBulkWarn = TRUE in this function. When running in a non interactive environment like CRON, the bulk request will be processed without a warning regardless.
fmpc_symbols_available

Examples

## Not run:
# Set the FMP Token. The DEMO token has VERY limited access.
fmpc_set_token('demo', timeBtwnReq = 0, noBulkWarn = TRUE)

## End(Not run)

fmpc_symbols_available

Get symbols available through FMP Cloud that have a profile

Description

Use a valid API Token to pull all symbols that have a profile on FMP Cloud. Pull the profile using fmpc_security_profile. Set API token using fmpc_set_token.

Usage

fmpc_symbols_available()

Value

a data frame of symbols with descriptions

Examples

## Not run:
# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
fmpc_symbols_available()

## End(Not run)

fmpc_symbols_by_market

Get available symbols across markets

Description

Pulls all symbols available on FMP Cloud that has historical pricing data. Markets can include equities, mutual funds, commodities, indexes, and more.
Usage

```r
fmpc_symbols_by_market(market = "all")
```

Arguments

```r
market
```
select one or more markets to pull available symbols. options include: 'all', 'etf', 'commodity', 'euronext', 'nyse', 'amex', 'tsx', 'index', 'mutual_fund', 'nasdaq', 'crypto', 'forex'

Value

a data frame of pricing, split, and dividend history

Warning

Running 'all' generates 11 API calls

Examples

```r
## Not run:
# A valid token must be set to use this function
fmpc_set_token("FMPAPIKEY")
fmpc_symbols_by_market(market = c("index", "commodity"))
fmpc_symbols_by_market() # default will pull all markets

## End(Not run)
```

---

`fmpc_symbols_index` Current or historical constituents for a specific index

**Description**

Shows current or historical companies in the S&P 500, Nasdaq, or Dow Jones

Usage

```r
fmpc_symbols_index(
    period = c("current", "historical"),
    index = c("sp500", "nasdaq", "dowjones")
)
```

Arguments

```r
period
```
'current' for current list, 'historical' for a list of companies that have been added and the ones that were replaced

```r
index
```
indicate the index to pull for: sp500, dowjones, nasdaq
Value

data frame of constituents

Examples

## Not run:

# Must set a valid API token
fmpc_set_token('FMPAPIKEY')
fmpc_symbols_index()
fmpc_symbols_index('historical','nasdaq')

## End(Not run)

fmpc_symbol_search  Search for symbols or companies

Description

Enter search string to pull back matching symbols or companies

Usage

fmpc_symbol_search(query = "apple inc", limit = 10)

Arguments

query  a search string, for example 'tech'
limit  limit the results to a certain number

Value

list of symbols and companies that meet the search criteria

Examples

## Not run:

# Function can work without a valid API token
fmpc_set_token()  # defaults to 'demo'
fmpc_symbol_search('apple')
fmpc_symbol_search('tech', 100)

## End(Not run)
Index

fmpc_13f_cik_list, 2, 5
fmpc_13f_cik_name, 3
fmpc_13f_cik_search, 4
fmpc_13f_data, 4
fmpc_analyst_outlook, 5
fmpc_calendar_events, 6
fmpc_cots_data, 7
fmpc_cots_symbols, 7, 8
fmpc_cusip_search, 9
fmpc_earning_call_transcript, 9
fmpc_economic_events, 10, 11
fmpc_economic_results, 10, 11
fmpc_financial_bs_is_cf, 12
fmpc_financial_dcfv, 13
fmpc_financial_metrics, 14
fmpc_financial_zip, 15
fmpc_get_url, 16
fmpc_held_by_mfs, 17, 18
fmpc_holdings_etf, 17, 18
fmpc_market_hours, 19
fmpc_price_batch_eod, 19
fmpc_price_current, 20
fmpc_price_forex, 21
fmpc_price_full_market, 21
fmpc_price_history, 22, 23
fmpc_price_history_spdiv, 23
fmpc_price_intraday, 24
fmpc_rss_sec, 25
fmpc_security_delisted, 26
fmpc_security_dividends, 23, 27
fmpc_security_gla, 28
fmpc_security_mrktcap, 28
fmpc_security_news, 29
fmpc_security_profile, 30, 37
fmpc_security_ratings, 31
fmpc_security_screener, 32
fmpc_security_sector, 33
fmpc_security_splits, 23, 34
fmpc_security_tech_indic, 35
fmpc_set_token, 6, 10, 17, 20–23, 25, 27, 29–31, 34, 35, 36, 37
fmpc_symbol_search, 39
fmpc_symbols_available, 6, 10, 29–31, 37
fmpc_symbols_by_market, 17, 20, 22–25, 27, 34, 35, 37
fmpc_symbols_index, 38