

Moduł pobierania danych

dokumentacja

Opis zmiennych globalnych:

REMBER_COUNT – liczba obiektów finansowych zapamiętywana w historii

DATABASE_LAST_UPDATE – zmienna przechowująca datę ostatniej aktualizacji bazy danych

INDEX_LIST – zmienna przechowująca listę indeksów giełdowych obsługiwanych przez aplikację (lista ta jest wczytywana z pliku przy każdym uruchomieniu aplikacji)

STOCK_LIST - zmienna przechowująca listę spółek giełdowych obsługiwanych przez aplikację (lista ta jest wczytywana z pliku przy każdym uruchomieniu aplikacji)

FOREX_LIST - zmienna przechowująca listę walut obsługiwanych przez aplikację (lista ta jest wczytywana z pliku przy każdym uruchomieniu aplikacji)

RESOURCE_LIST - zmienna przechowująca listę surowców obsługiwanych przez aplikację (lista ta jest wczytywana z pliku przy każdym uruchomieniu aplikacji)

BOND_LIST - zmienna przechowująca listę obligacji obsługiwanych przez aplikację (lista ta jest wczytywana z pliku przy każdym uruchomieniu aplikacji)

FUTURES_LIST - zmienna przechowująca listę kontraktów terminowych obsługiwanych przez aplikację (lista ta jest wczytywana z pliku przy każdym uruchomieniu aplikacji)

Elementy powyższych list są postaci:
[skrót, nazwa_elementu, źródło, opis]

HISTORY_LIST – zawiera listę obiektów typu FinancialObject, jej długość ograniczona jest przez zmienną REMEMBER_COUNT. Zawartość tej listy serializowana jest do pliku data.wsf przy każdym zamknięciu aplikacji i wczytywana przy jej uruchomieniu.

AMEX_HIST – lista zawierająca dane historyczne na temat ilości spółek rosnących i spadających z indeksu Amex.

NYSE_HIST – lista zawierająca dane historyczne na temat ilości spółek rosnących i spadających z indeksu NYSE.

NASDAQ_HIST – lista zawierająca dane historyczne na temat ilości spółek rosnących i spadających z indeksu Nasdaq.

Elementy powyższych list są postaci: [data, ilość rosnących, ilość spadających, ilość stałych, wolumen rosnących, wolumen spadających, wolumen stałych]

UPDATE_FLAG – flaga służąca do określenia czy obiekt finansowy jest aktualizowany czy pobierany jest nowy obiekt.

Opisy klas:

FinancialObject

klasa definiująca obiekt finansowy (index,spółkę,surowiec,obligację, etc.), w której przechowywane są aktualne jak i archiwalne notowania.

Pola:

- name – pełna nazwa obiektu
- abbreviation – skrót obiektu
- financialType – rodzaj obiektu finansowego(forex, index etc.)
- dataSource – źródło, skąd pobierane są dane (Yahoo/Stooq)
- detail – szczegół opisujący obiekt
- lastUpdate – data ostatniej aktualizacji obiektu
- currentValue – lista złożona z aktualnej ceny obiektu i czasu jej pobrania
- previousValues – lista historycznych wartości z danego dnia
- valuesDaily – lista historycznych dziennych wartości
- valuesWeekly – lista historycznych tygodniowych wartości
- valuesMonthly – lista historycznych wartości miesięcznych
- dailyUpdate – data ostatniej historycznej wartości dziennej
- weeklyUpdate – data ostatniej historycznej wartości tygodniowej
- monthlyUpdate – data ostatniej historycznej wartości miesięcznej

Konstruktory:

FinancialObject(name, abbreviation, financialType, dataSource, detail = None,lastUpdate = datetime.date(1971,1,1))

Parametry:

- name – pełna nazwa obiektu.
- abbreviation – skrót obiektu.
- financialType – rodzaj obiektu finansowego(forex, index etc.)
- dataSource – źródło, skąd pobierane są dane (Yahoo/Stooq)
- detail – szczegół opisujący obiekt
- lastUpdate – data ostatniej aktualizacji obiektu

Opis:

Tworzy obiekt klasy o zadanych parametrach.

Metody:

`getCurrentValue()`

Parametry:

Opis:

Metoda aktualizująca dane dotyczące aktualnej wartości obiektu oraz przenosząca poprzednią wartość do listy poprzednich wartości

`updateArchive(timePeriod)`

Parametry:

`timePeriod` – String określający częstość pobieranych danych(daily, weekly, monthly)

Opis:

Metoda aktualizująca dane istniejącego obiektu. Tworzy nowy tymczasowy obiekt i kopiuje jego zawartość do obiektu 'self'. Aktualizowany jest przedział czasowy podany w parametrze. Wartości pobierane są od ostatniej aktualizacji do dnia wczorajszego. Rzuca wyjątek `DataAPIException`.

`getArray(time)`

Parametry:

`time` – String określający, którą listę wartości chcemy przekształcić w tablicę(daily, weekly, monthly)

Opis:

Funkcja zwracająca rekordowaną tablicę (`numpy.recarray`) dla informacji w odstępie czasu przekazanym jako parametr funkcji. Pozwala to dostać się do poszczególnych tablic używając odpowiednich rekordów: 'date' 'open' etc.

getIndex(begin, end, time)

Parametry:

begin – data rozpoczynająca przedział czasowy

end – data kończąca przedział czasowy

time - string określający dla której tabeli wartości szukamy przedziału czasu (daily, weekly, monthly)

Opis:

Funkcja zwracająca indeksy tablicy dla danego przedziału czasu.

DataAPIException(Exception)

Wyjątek zwracany przez moduł pobierania danych.

Konstruktory:

DataAPIException(value)

Parametry:

value – opis, który wystąpił.

Opis:

Wyjątek rzucany przez moduł pobierania danych z opisem błędu jako parametrem wywołania.

Opisy funkcji:

createWithCurrentValueFromYahoo(name, abbreviation, financialType, detail)

Parametry:

name – nazwa obiektu finansowego.

abbreviation – skrót nazwy obiektu(eg. Apple Inc. → AAPL).

financialType – rodzaj obiektu finansowego(eg. „Index”, „Stock”).

detail – krótki opis obiektu lub jego szczególna cecha.

Opis:

Funkcja tworząca nowy obiekt FinancialType (lub aktualizująca istniejący jeśli znajduje się on w HISTORY_LIST) z aktualną wartością spółki, której szczegóły zostały przekazane w parametrach funkcji. Dane spółki pobierane są ze strony finance.yahoo.com na zasadzie pobrania zawartości odpowiedniej strony i wyszukania za pomocą wyrażeń regularnych interesujących nas liczb.

`createWithCurrentValueFromStooq(name, abbreviation, financialType, detail)`

Parametry:

- `name` – nazwa obiektu finansowego.
- `abbreviation` – skrót nazwy obiektu(eg. Apple Inc. → AAPL).
- `financialType` – rodzaj obiektu finansowego(eg. „Index”, „Stock”).
- `detail` – krótki opis obiektu lub jego szczególna cecha.

Opis:

Funkcja tworząca nowy obiekt `FinancialType` (lub aktualizująca istniejący jeśli znajduje się on w `HISTORY_LIST`) z aktualną wartością obiektu, którego szczegóły zostały przekazane w parametrach funkcji. Dane spółki pobierane są ze strony `stooq.pl` na zasadzie pobrania zawartości odpowiedniej strony i wyszukania za pomocą wyrażeń regularnych interesujących nas liczb.

`createWithArchivesFromYahoo(name, abbreviation, financialType, detail, timePeriod, sinceDate = datetime.date(1971,1,1))`

Parametry:

- `name` – nazwa obiektu finansowego.
- `abbreviation` – skrót nazwy obiektu(eg. Apple Inc. → AAPL).
- `financialType` – rodzaj obiektu finansowego(eg. „Index”, „Stock”).
- `detail` – krótki opis obiektu lub jego szczególna cecha.
- `timePeriod` – string określający częstość pobieranych danych(daily, weekly, monthly)
- `sinceDate` – data, od której pobierane będą dane archiwalne, domyślnie ustawiona na 1-1-1971

Opis:

Funkcja zwracająca nowy obiekt `FinancialType` (lub aktualizująca istniejący jeśli znajduje się on w `HISTORY_LIST`), który zawiera archiwalne wartości od podanej daty do dnia obecnego dla obiektu, którego szczegóły parametrami funkcji. Dane spółki pobierane są ze strony `finance.yahoo.com` na zasadzie pobrania odpowiedniego pliku csv i wczytywaniu wartości wiersz po wierszu.

`createWithArchivesFromStooq(name, abbreviation, financialType, detail, timePeriod, sinceDate = datetime.date(1971,1,1))`

Parametry:

name – nazwa obiektu finansowego.

abbreviation – skrót nazwy obiektu(eg. Apple Inc. → AAPL).

financialType – rodzaj obiektu finansowego(eg. „Index”, „Stock”).

detail – krótki opis obiektu lub jego szczególna cecha.

timePeriod – string określający częstość pobieranych danych(daily, weekly, monthly)

sinceDate – data, od której pobierane będą dane archiwalne, domyślnie ustawiona na 1-1-1971

Opis:

Funkcja zwracająca nowy obiekt FinancialType (lub aktualizująca istniejący jeśli znajduje się on w HISTORY_LIST), który zawiera archiwalne wartości od podanej daty do dnia obecnego dla obiektu, którego szczegóły parametrach funkcji. dane spółki pobierane są ze strony stooq.pl na zasadzie, że najpierw pobierane są cookies potrzebne do nawiązania połączenia a następnie pobierany jest odpowiedni plik csv i wczytywane są wartości wiersz po wierszu.

`parserStringToDate(string)`

Parametry:

string – data w formacie „YYYY-MM-DD”

Opis:

Funkcja zmieniająca ciąg znaków postaci "YYYY-MM-DD" na obiekt klasy datetime.date

`parserDateToString(date)`

Parametry:

date – obiekt klasy datetime.date

Opis:

Funkcja zamieniająca obiekt klasy datetime.date na string postaci YYYYMMDD

updateDatabase()

Parametry:

Opis:

Funkcja aktualizująca plik data1.wsf o spółki IPO debiutujące na amerykańskiej giełdzie. Funkcja pobiera nazwy i skróty debiutujących spółek od miesiąca ostatniej aktualizacji do czasów obecnych i jeśli nie ma ich już w bazie danych to są one dodawane. (Dane pobierane są ze strony biz.yahoo.com/ipo)

loadData()

Parametry:

Opis:

Funkcja wczytująca bazę danych z plików do globalnych list. Wczytywana jest lista indeksów, spółek, walut, surowców, obligacji i kontraktów. Wczytywane są także dane na temat ilości spółek rosnących i spadających poszczególnych indeksów w poszczególnych dniach.

getAdvDec(date)

Parametry:

date – obiekt klasy datetime.date określający dla jakiego dnia chcemy pobrać dane.

Opis:

Funkcja, która pobiera dane i na temat ilości spółek rosnących i spadających, a także na temat ich wolumenu z dnia przekazanego w parametrze funkcji i zapisuje je do odpowiednich plików. Dane pobierane są z tabel umieszczonych na stronie <http://unicorn.us.com/advdec/>

updateAdvDec()

Parametry:

Opis:

Funkcja aktualizuje pliki NYSE.wsf, AMEX.wsf, NASDAQ.wsf wywołując funkcje getAdvDec(date) dla każdej daty od ostatniej aktualizacji do dnia obecnego.

getAdvDecInPeriodOfTime(begin,end,index)

Parametry:

begin – początek zakresu(datetime.date).

end – koniec zakresu(datetime.date).

index – nazwa indeksu (AMEX,NYSE,NASDAQ), którego dane chcemy pobrać.

Opis:

Funkcja zwraca etykietowaną tablicę numpy.array postaci:

[('date','S10'),('adv',int),('dec',int),('unc',int),('advv',int),('decv',int),('uncv',int)]. W tablicy tej znajdują się dane pobrane z odpowiedniej globalnej listy.

isInHistory(abbreviation)

Parametry:

abbreviation – skrót obiektu, którego szukamy.

Opis:

Funkcja sprawdzająca, czy obiekt o podanym skrócie znajduje się już w historii. Jeśli tak to zwracany jest znaleziony obiekt jeśli nie otrzymujemy wartość “none”

isInStock(abbreviation)

Parametry:

abbreviation – skrót spółki, której szukamy.

Opis:

Funkcja sprawdza czy dana spółka znajduje się już w bazie danych.

saveHistory(file):

Parametry:

file – plik, do którego chcemy zapisać historię.

Opis:

Funkcja serializuje za pomocą biblioteki cPickle listę History_List do pliku podanego w parametrze funkcji.

loadHistory(threading.Thread):

Parametry:

file – plik, z którego wczytywana będzie historia.

Opis:

Klasa - wątek wczytująca zserializowaną listę HISTORY_LIST z pliku.

top5Volume()

Parametry:

Opis:

Funkcja pobiera nazwę, wartość, %zmiany dla 5 spółek o największym aktualnie wolumenie z yahoo.finance.

top5Gainers()

Parametry:

Opis:

Funkcja pobiera nazwę, wartość, %zmiany dla 5 spółek o największym aktualnie %wzroście z yahoo.finance.

top5Losers()

Parametry:

Opis:

Funkcja pobiera nazwę, wartość, %zmiany dla 5 spółek o największej aktualnie %stracie z yahoo.finance.

getMostPopular()

Parametry:

Opis:

Funkcja zwracająca wartości najbardziej popularnych obiektu.