Bitcoin Trader

Date	23 March 2016
Author	Robert Hoff
Contact	hoff.rj@gmail.com
Pages	6

1 Source files

Code volume, 5000 lines

📁 **main**

Main runtime, entry points, global utilities and loggers

Class-name	
AppMain	Runtime entry for the trader, configurable to exchanges under observation, strategy and trade-limits
AppChart	Entry for visualisations and file-exports of graphs, comparisons, filters, and trade-simulations
LoggerConfig	Configuration for logger settings, definitions for production, development and testing, adaptable to custom log-levels and exchanges
Timer	Helper for creating timestep boundaries, conversion of UTC string representations to UTC int and long timestamps. Strings are accepted at different resolutions E.g. "2015-01-01", "2015-01-01 10:00", and "2015-01-01 10:00:30"

📁 **chart**

Chart visualisation and file-exporting

Class-name	
DataCharting	Entry for all chart-definitions, accepts calls corresponding to analytical intents (represented by plot-types). Provides configuration for labelling, axis and layout, and provides methods for canvas rendering and file exporting (single or batched over intervals)
TradePlot	Parent class for all plot-types, provides axis-ranges and time-scales (Second, Hour or Day)
SeriesPlot	Parent class for all data that can be plotted linearly (observed by trade-series, timestep, shift-step, ave, shift-ave, ave-normalised, and mean-ind)
Mean3Plot	A special time-series plot that compares relative mean of three exchanges under observation
VolumePlot	Volume plot that can be appended to trade-series or timestep data
VolumePlotXYItem	Helper object for VolumePlot that determines the graphical extent of the vertical bars
VolumePlotMouseListener	Event listener for VolumePlot that shows individual volume data as tooltips for a given bar, also provides data dumps on all trades in a given interval (responding to mouse-click)

📁 chart/series

Analytical methods that can be plotted by SeriesPlot

Class-name	
TradeSeries	Parent class for all series data
SeriesTimeStep	Timestep series averaged over time-increments in seconds. "new SeriesTimeStep(0)" is equivalent to plotting all trades over given interval, "new SeriesTimeStep(1)" will average on 1 second intervals, *etc*
SeriesMovAve	Implements a moving average, linearly scaled from a central point
Additional, briefly	
SeriesShiftStep	
SeriesShiftAve	
SeriesShiftAveNormalised	
SeriesShiftMeanInd	

📁 chart/data

Data processing classes for series data

Class-name	
TradeInterval	
TradeSet	
TradeSizeDistribution	
XYData	
MovingAve	
MeanReference	
MeanIndicator	

📁 data/importing

Classes for importing historical data, either from CSV files (such as api.bitcoincharts.com/v1/csv/), or from exchange APIs

Class-name	
FetchAPItrades	Collect historical trades from given exchange API
ProcessCSVData	Populate database with trade-data provided as CSV files, given as data triplets (timestamp, price, amount)
ParseCSVTriple	Auxiliary classes needed for the CSV-processing
TradeDataTriple	

📁 data/fx-service

Classes for updating and retrieving historical and current exchange rates (taken from third-party provider)

Class-name	
FxData	Interface for providing exchange rate on a currency "base-symbol" on historical or current timestamp. Symbols supported are EUR, GBP, CNY, PLN, RUB, CAD, AUD, HKD, CHF, JPY, MXN, BRL, ZAR, ILS, NOK
FxDataService	API wrapper to access live FX rates on base-symbols above

📁 trader

Classes that run the live trader and implements trading strategies

Class-name	
Trader	Live trader that executes trades according to provided strategy
ExchangeState	Comparative state representations of exchanges under observation that provide data, and data-analysis, that the trader relies on
Trading-strategies	
[...]	Trading strategies implemented as sub-classes of Trader

📁 exchApi

API wrappers that provide an application consistent view of exchanges under observation.
Exchanges enabled are Anx, Bistamp, Btce, Btcn, Cexio, Crypsy, HitBtc, Itbit, LakeBtc, OkCoin.

Class-name	
ExchModel	Adapted class that provides consistent functionaliy and modeling across API differences, access point for updates on ticker, market depth, proprietary open orders, proprietary trade-history and proprietary balance, and interface for dispatching and cancelling trading instructions
ExchAPI	Parent class for all exchange-API
API names	
ExchangeApiAnx	
ExchangeApiBitstamp	
ExchangeApiBtce	
ExchangeApiBtcn	
ExchangeApiCexio	
ExchangeApiCrypsy	
ExchangeApiHitBtc	
ExchangeApiItBtc	
ExchangeApiLateBtc	
ExchangeApiOkCoin	

📁 **models**

Models for the database ORM (using MySQL)

Class-name	
Exchange	Application-specific model for each of the exchanges
Balance	Record of proprietary balance
OpenOrder	Record of proprietary open-orders (marked as open, fulfilled or cancelled)
TradeProp	Record of proprietary executed trades
Request	Record of all requests made on exchange APIs
ExceptionLog	Record of all runtime exceptions, caught by AppMain
FxRate	Record of FX-rates on relevant base-symbols. (Recorded once every day only, historical rates are returned as linear extrapolations)
Trade	Record of historic trades (on high volume exchanges averaged on timestep(1), limits intra-day records to 86400)
Ticker	Selected record of historical tickers (sporadic, most analysis does not depend on them)

2 Data-exporting example

Comparison between exchanges Bitstamp (European based) and Btce (Russian based), intra-day, volumes over 12 minute intervals.

27 July 2015

Exchange	Nr of trades	Intra-day volume (total)
Bitstamp	4018	4.27 million USD
Btce	5028	1.78 million USD

L-Ave (200) denotes 200 second moving average, applied to both trade series. All axis in USD not converted from foreign currencies.

bitstamp-btce 2015-07-27