A document containing the

Bully Stock Exchange JDBC API

Joseph Langley

Current Planned Functionality

- Get stock quotes
- Add and delete user accounts
- Record stock trading activity
- Manage company information

Method Requirements per Function

Get Stock Quotes

- Vector GetQuote(String symbol);
  Returns a vector containing vectors of string pairs in the following format
  (("Symbol", Symbol),
  ("Open", Open),
  ("Last", Last),
  ("Volume", Volume),
  ("P/E", p_e_ratio),
  ("Change", change),
  ("%Change", p_change),
  ("Yield", yield),
  ("High", high),
  ("Low", low))

  Note: The order of this vector is not yet fixed.

- String doLogon(String username, String password);
  Returns AccountId from database if username and password match, otherwise returns (String)null.

- String checkLogonCookie();
  Gets accountId from local cookie if a successfull logon has occured. If timeout has occured, calls logon (in an applet window?). Returns accountId if successfull. Returns (String)null if timeout has occured and logon fails.

- String setLogonCookie(String username, String password);
  Sets local cookie with accountId if username and password matches database. Calls doLogon(String username, String password).

- Vector GetHoldings(String accountId);
  Returns a vector of vectors containing information about the user's portfolio:
  (("Cash", bullyse.Portfolio.Balance),
  (Symbol, NumShares, BoughtAt, Current, Profit/Loss)
  -
  -
  -)

Note: The first element is a string pair representing the user's available, liquid funds. The remaining elements are vectors which represent the user's commodities.

- Vector GetHoldings(String accountId, String symbol);
  Returns a commodity vector (as for the previous overload) if the user owns any shares of the stock referenced by 'symbol.' If the user owns no shares of the stock, GetHoldings() returns (Vector)null.

- double CalculateWealth(String accountId);
  Returns a double that represents the total wealth present in the BullySE system. Calculated as bullyse.Portfolio.Balance plus Last for all owned stocks times number of shares owned.
  Wealth = Balance + sum(Last(NumShares))
- boolean BuyStock(String symbol, String accountId, int numShares);
  Searches the available stocks specified and tries to buy the specified number of shares. Fails and returns false if:
  - User doesn't have enough cash
  - numShares excedes available and user doesn't allow partial fulfillment
  - symbol doesn't exist
  - the required tables can't be locked.

  Otherwise the cost of the stock is deducted from the user's Balance, the stocks are added to the user's Holdings, the stocks are subtracted from the available list, Last is updated in Companies, funds are added to the Balance of the selling parties, the trade is added to Trades (for both buying and selling parties) and the appropriate HistoryTable, and the method returns true.

- boolean SellStock(String symbol, String accountId, int numShares);
  Tries to sell stocks according to algorithms to be defined later.
  Fails and returns false if:
  - required tables can't be locked
  - symbol doesn't exist
  - user doesn't own any of the stock
  - numShares exceeds user's holdings
  - Other bad conditions occur.

  Otherwise the stocks are placed on the available list, stocks are deducted from user's Holdings, and the method returns true.

- Vector GetPricePoints(String symbol);
  Returns a vector of the price points for the stock referenced by 'symbol.'
  The vector will contain some formatting information as a vector in the first element.
  This is not a pressing objective.

- Vector GetMarketAvg();
  Returns price points representing the market average.
  See GetPricePoints() above.

Example of calling GetHoldings()

GetHoldings(checkLogonCookie(), symbol);