

API Documentation

API Documentation

March 12, 2013

Contents

Contents	1
1 Module moira	2
1.1 Functions	2
1.2 Variables	3
1.3 Class Portfolio	4
1.3.1 Methods	4
1.4 Class Stock	4
1.4.1 Methods	4
1.5 Class Trans	5
1.5.1 Methods	5
Index	6

1 Module moira

MOIRA, the MOIRA Otto-matic Intelligent Reconitter of Assets, is an API for the Marketwatch Virtual Stock Exchange game.

Code is available on Github¹.

1.1 Functions

get_current_holdings(*token, game*)

Fetches and parses current holdings.

Parameters

token: Cookiejar returned by a call to **get_token**.

game: The *name* of the game (marketwatch.com/game/XXXXXXX).

Return Value

Stock data.

(*type=Dict of **Stock** objects, keyed by id*)

Warning: The stock price returned by a call to **get_current_holdings** is rounded to the nearest cent! This results in inaccuracies if you calculate things based on this number — don't. Use **stock_search** instead. Interestingly, Marketwatch itself never reports the full-precision stock price anywhere except in HTML attributes.

get_portfolio_data(*token, game*)

get_token(*username, password*)

Issues a login request. The token returned by this function is required for all methods in this module.

Parameters

username: The marketwatch.com username (email).

password: The plaintext marketwatch.com password.

Return Value

Requests cookiejar containing authentication token.

Note: It's unknown what the expiry time for this token is - it is set to expire at end of session. It may be apt to request a new token daily, while the market is closed.

¹<http://github.com/brandonwu/moira>

get_transaction_history(*token, game*)

Downloads and parses the list of past transactions.

Parameters

token: Cookiejar returned by `get_token`.

game: The *name* of the game (marketwatch.com/game/XXXXXXX).

Return Value

A dict of all past transactions.

(*type=Dict of Trans objects, keyed on an index (1, 2...).*)

order(*token, game, type, id, amt*)

Initiates a buy, sell, short, or cover order.

Parameters

token: Cookiejar returned by `get_token`.

game: Game name (marketwatch.com/game/XXXXXXX)

id: Security ID (not the ticker symbol). Obtain from `stock_search`

amt: Order amount.

type: Type of order - 'Sell', 'Buy', 'Short', or 'Cover'.

Return Value

Returns integer - 0 if success, nonzero if failure.

(*type=integer*)

stock_search(*token, game, ticker*)

Queries Marketwatch for stock price and ID of a given ticker symbol.

Parameters

token: Cookiejar returned by `get_token`.

game: Game name (marketwatch.com/game/XXXXXXX).

ticker: Ticker symbol of stock to query.

Return Value

Current stock price, stock id, and server time.

(*type=Dict {'price':float, 'symbol':str, 'time':datetime object in EST}.*)

Note: You must have joined a game in order to use this function.

1.2 Variables

Name	Description
<code>__package__</code>	Value: None
<code>ch</code>	Value: <logging.StreamHandler object at 0x1a57990>
<code>formatter</code>	Value: <logging.Formatter object at 0x1aca4d0>
<code>from_zone</code>	Value: tzfile('/usr/share/zoneinfo/UTC')
<code>logger</code>	Value: <logging.Logger object at 0x1a4ae90>

continued on next page

Name	Description
to_zone	Value: tzfile('/usr/share/zoneinfo/America/New_York')

1.3 Class Portfolio

Stores portfolio data.

1.3.1 Methods

__init__ (<i>self, time, cash, leverage, net_worth, purchasing_power, starting_cash, return_amt</i>)	
Parameters	
time:	Last updated time (server time from HTTP headers).
cash:	Amount of <i>cash</i> (not purchasing power!) remaining.
leverage:	Amount available to borrow.
net_worth:	Sum of assets and liabilities.
purchasing_power:	Amount (credit + cash) available to buy.
starting_cash:	Cash amount provided at game start.
return_amt:	Dollar amount of returns over starting_cash .

1.4 Class Stock

Stores portfolio data for a single stock.

1.4.1 Methods

__init__ (<i>self, id, ticker, security_type, current_price, shares, purchase_type, returns</i>)	
Parameters	
id:	Unique ID assigned by Marketwatch to each security.
ticker:	The ticker symbol of the stock.
security_type:	"ExchangeTradedFund" or "Stock"
current_price:	Current price per share, <i>rounded to the cent</i> .
shares:	Number of shares held.
purchase_type:	"Buy" or "Short"
returns:	Total return on your investment.
purchase_price:	Price at time of purchase. @see See the warnings at get_current_holdings about price rounding.

1.5 Class Trans

Stores transaction data for a single transaction.

1.5.1 Methods

```
__init__(self, ticker, order_time, trans_time, trans_type, trans_amt, exec_price)
```

Parameters

ticker: The ticker symbol of the security.
order_time: The time the order was issued.
trans_time: The time the order was executed.
trans_type: "Buy", "Short", "Sell", or "Cover"
trans_amt: Number of shares sold/purchased.
exec_price: Price of security at time of order.

Index

id, 2

moira (*module*), 2–5

 moira.get_current_holdings (*function*), 2

 moira.get_portfolio_data (*function*), 2

 moira.get_token (*function*), 2

 moira.get_transaction_history (*function*), 2

 moira.order (*function*), 3

 moira.Portfolio (*class*), 4

 moira.Portfolio.__init__ (*method*), 4

 moira.Stock (*class*), 4

 moira.Stock.__init__ (*method*), 4

 moira.stock_search (*function*), 3

 moira.Trans (*class*), 4–5

 moira.Trans.__init__ (*method*), 5

name, 2, 3