

# API Documentation

API Documentation

March 11, 2013

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Module moira</b>	<b>2</b>
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 . . . . .	4
1.5.1 Methods . . . . .	5
<b>Index</b>	<b>6</b>

# 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 Github<sup>1</sup>.

## 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.

---

<sup>1</sup><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...).*)

**sell\_stock**(*token, game, id, amt*)

Initiates a sell 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:** Amount of stock to sell.

**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>	<b>Value:</b> None
<code>ch</code>	<b>Value:</b> <logging.StreamHandler object at 0x13aa950>
<code>formatter</code>	<b>Value:</b> <logging.Formatter object at 0x141d490>
<code>from_zone</code>	<b>Value:</b> tzfile('/usr/share/zoneinfo/UTC')
<code>logger</code>	<b>Value:</b> <logging.Logger object at 0x139de50>
<code>to_zone</code>	<b>Value:</b> tzfile('/usr/share/zoneinfo/America/New_York')

## 1.3 Class Portfolio

Stores portfolio data.

### 1.3.1 Methods

```
__init__(self, time, cash, leverage, net_worth, purchasing_power, starting_cash, return_amt)
```

**Parameters**

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

## 1.4 Class Stock

Stores portfolio data for a single stock.

### 1.4.1 Methods

```
__init__(self, id, ticker, security_type, current_price, shares, purchase_type, returns)
```

**Parameters**

<b>id:</b>	Unique ID assigned by Marketwatch to each security.
<b>ticker:</b>	The ticker symbol of the stock.
<b>security_type:</b>	"ExchangeTradedFund" or "Stock"
<b>current_price:</b>	Current price per share, <i>rounded to the cent</i> .
<b>shares:</b>	Number of shares held.
<b>purchase_type:</b>	"Buy" or "Short"
<b>returns:</b>	Total return on your investment.
<b>purchase_price:</b>	Price at time of purchase. @see See the warnings at <b>get_current_holdings</b> 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.Portfolio (*class*), 3–4

        moira.Portfolio.\_\_init\_\_ (*method*), 4

    moira.sell\_stock (*function*), 3

    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