Backend Task

Portfolio API

A portfolio tracking API which allows adding/deleting/updating trades and can do basic return calculations etc.
For simplicity assume that there will be only one portfolio and one user.

The portfolio is essentially a collection of stocks, each stock having multiple trades (buy/sell). Each trade can have only one stock, a quantity and a date.

Example portfolio:

RELIANCE:

BUY 100@900 10/04/2015
SELL 50@1000 10/05/2015
BUY 100@850 10/06/2015

HDFCBANK:

BUY 200@1000 11/05/2015
SELL 100 @800 12/07/2015

Holdings:
RELIANCE: 150 @ 875.5 (Avg of all buys)
HDFCBANK: 100 @ 1000

API

Models

- Stock - just an alphanumeric id. Just pick any random strings for now, you do not need to create the whole stock collection. Use yahoo finance if you're feeling adventurous
- Trade - Should capture a date, price, type (buy/sell)
- Portfolio - an aggregation over stocks and trades

Functionality

- Retrieve the portfolio
- Add/delete/modify trades
- Get the average buying price

Calculate the average buying price as the average of all buys disregarding sells.

Routes

Route		Method	Parameters	Response
/portfolio	/	GET	-	Return the entire portfolio with trades
	/holdings	GET	-	Get holdings in an aggregate view (average buying price)
	/addTrade	POST	trade	-
	/updateTrade	POST	trade	-
	/removeTrade	POST	trade	-

This is just a guideline; you are free to make changes/additions as long as the basic functionality is captured.

Expected response (JSON) -

```
{
    success: Boolean,
    data: object
}
```

We would prefer if you use Python, MySQL. Other databases are fine if you can think of compelling reasons.

Suggested: Python Flask framework, POSTMAN for testing

Write code as you normally would. Things to remember -

- Modular code looks good
- Comments help

If you are just getting started with Python, have a look at scotch.io tutorials, or search YouTube for getting started with Python-Flask stack.

Feel free to use any additional tools that help you with the task or you feel would improve the overall app. Also, please work on a brief write up detailing design decisions, tests you would run and possible improvements you would have made.