

## Exercise

### Description:

It is 1999 and a local metal band is looking to have a bake sale to raise funds for their new album. They have promised you a free CD if you make software to help run the sale; they are musicians not mathematicians after all.

There are four items they would like to sell on this sale with specific prices and quantities of each:

Brownie - 65c (48)

Muffin - \$1.00 (36)

Cake Pop - \$1.35 (24)

Water - \$1.50 (30)

The application must calculate the smallest amount of change to give a person if they overpay. If you do not have stock of an item, you cannot make the sale.

Purchases are input as comma delimited string. The first letter of each item is its input code, B for Brownie, M for Muffin, C for Cake Pop and W for Water.

The application then responds with a total if there are enough of each item.

Otherwise responds with **"Not enough stock."**

If the total appears the user, then inputs the amount paid. If the amount is equal to or greater than the amount due the application, then figures out the amount of change to give.

If the amount is less than the amount due it responds with **"Not enough money."**

### Technical stuff:

Build a neat front-end and backend. Use REST and database you see best suited for the assignment.

### Examples:

Items to Purchase > B, C, W Total > \$3.50 Amount Paid > \$4.00 Change > \$0.50	Items to Purchase > B Total > \$0.65 Amount Paid > \$0.75 Change > \$0.10	Items to Purchase > C,M Total > \$2.35 Amount Paid > \$2.00 Change > Not enough money	Items to Purchase > W Total > Not enough stock
--	--	---	---

### Hint:

Looks to use Stubs or Mocks or both when solving input and output problems.  
Use clean code principles and test driven development.

### Bonus:

Add functionality to read the items, quantity and price all from a configuration file to database.