Developer Recruitment

Tech Test #1 - Checkout

Version 1.1

Author: Martin Mayer

Date: 20 August 2019

Contents

[Developer - Checkout Kata 4](#_Toc54270897)

[Problem Synopsis 4](#_Toc54270898)

[Format 4](#_Toc54270899)

[Deliverables 4](#_Toc54270900)

[Kata Review 5](#_Toc54270901)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Description | Date | Author |
| 0.1 | Initial draft | 05/08/2019 | M Mayer |
| 1.0 | Removed Draft watermark | 06/08/2019 | M Mayer |
| 1.1 | Added in local git zip as alt to public repo | 20/08/2019 | M Mayer |
| 1.2 | Removing Example Starting Point | 22/10/2020 | E Smith |

# Developer - Checkout Kata

## Problem Synopsis

A supermarket requires a working checkout. MVP is to scan products and periodically ask for a total price, considering any special offers that apply to the product.

Items:

|  |  |
| --- | --- |
| **SKU** | **Unit Price** |
| A99 | 0.50 |
| B15 | 0.30 |
| C40 | 0.60 |

Special Offers:

|  |  |  |
| --- | --- | --- |
| **SKU** | **Quantity** | **Offer Price** |
| A99 | 3 | 1.30 |
| B15 | 2 | 0.45 |

The checkout accepts items scanned in any order, so that if we scan a pack of Biscuits (B15), an apple (A99) and another pack of biscuits, we’ll recognise two packs of biscuits and apply the discount of 2 for 45. Prove your solution works for this scenario.

## Format

Manage your time effectively to progress through the main points, do not get distracted on earlier deliverables (such as with very detailed testing). For the purposes of this short kata, showing the primary deliverables and talking about broader work is fine.

## Deliverables

Please push your work to a remote git repository (e.g. GitHub) and share the location and any required credentials with us. Commit as you go to show your working process, rather than just one big commit at the end. If you can’t share a public repository, please zip the local git repository and send that so we can still see your commits.

Work your way through this checklist:

* Create a new solution
  + Include a class library for the logic
  + Include a test library for unit tests (feel free to use whatever test library you are most comfortable with)
* Prove you can scan an item at a checkout
* Prove you can request the total price
* Introduce special offers
  + Amend your prior implementation to consider offers on items
* Prove you can request the total price inclusive of offers

This kata covers just the middleware, do not be concerned with UI or data access.

## Kata Review

Your solution will be measured against the deliverables above, and how you managed your time across them. No one solution is “correct”, but we are interested in:

* Your process
* Testable code
* Maintainable code
* Abstraction at sensible points
* How you would refactor your solution to deliver future requirements