Home Challenge

QikServe Engineer Technical Test

Thanks for your interest and taking the time to complete our technical test. This test consists of two parts:

1. Coding Challenge
2. Follow-up Questions

Submission Guidelines

Please submit a single zip file named {yourname}.zip containing the following:

1. One folder containing your solution to the Coding Challenge (including instructions on how to run your solution and any further relevant details)
2. A single markdown file with answers to the Follow-up Questions

Coding Challenge

Your challenge is to build a checkout system for a local supermarket. Think about how the solution would be used to calculate the total cost of a basket which could contain any combination of items and promotions. Please bear in mind the following:

- Items can be scanned in any order.
- Promotions should always be applied if possible.
- The supermarket owner would like to show the customer how much they've saved.
- New types of prices/promotions may need to be added in the future.

We have provided a WireMock server with a few API mappings. Please consume this API to retrieve production information for your application.

Note: Prices are expressed in pennies.

Requirements

Feel free to spend as much time on as you need on the challenge as long as the following requirements are met.

- Complete the user stories below
- Your code should be runnable in one step
- You must include tests
- Avoid including artefacts/build directories in your final zip file

User Stories

As a **user** running the application
I can **add items to the basket**
So I can **buy some items**

As a **user** running the application
I can **'checkout'** once I have finished adding items to the basket
So that I can **see the total price of the basket**

As a **user** running that application
I can see the **price details of my order** once I have checked out
So that **I can see how much I have spent and (potentially) saved**

Examples

Example 1

Basket Contents	Expected Totals	
1x Amazing Pizza!	Total :	£22.97
1x Amazing Burger!	Total Promos:	£0.00
1x Boring Fries!	Total Payable:	£22.97

Example 2

Basket Contents	Expected Totals	
1x Amazing Pizza!	Raw Total:	£27.96
1x Amazing Burger!	Total Promos:	£0.49
1x Amazing Salad!	Total Payable:	£27.47
1x Boring Fries!		

Example 3

Basket Contents	Expected Totals	
2x Amazing Pizza!	Raw Total:	£23.97
1x Boring Fries!	Total Promos:	£3.99
	Total Payable:	£19.98

Example 4

Basket Contents	Expected Totals	
3x Amazing Burger!	Raw Total:	£34.96
1x Amazing Salad!	Total Promos:	£10.48
	Total Payable:	£24.48

Note: You can visualise the basket totals any way you wish in you solution: command-line output, custom UI, API responses etc. However, please make sure you show how much the user has saved for each applicable promotion (if any).

Platform Choice

You can create the application in whatever way you feel best, for example: Command line application, Web App, Web API, through a main runner method or by running a sequence of automated tests. Please feel free to write your application in one of the following languages:

- Java
- C#
- JavaScript

Feel free to use any frameworks/libraries/packages you would like. This challenge is to examine your knowledge, reasoning and technical principals. We don't want to trick you or have you second guessing! Feel free to state any assumptions you've made, especially if anything we asked for is vague (sorry if that's the case 🙂). We are looking for something to showcase your awesome abilities. So clear code, good practices, something that you and your team mates would love.

Follow-up Questions

1. How long did you spend on the test?
2. What would you add if you had more time?
3. How would you improve the product APIs that you had to consume?
4. What did you find most difficult?
5. How did you find the overall experience, any feedback for us?

Thanks for your time and we look forward to hearing from you!

Inspiration for the test format taken with love from JustEat's recruitment test.