

Brief

For this exercise you will develop a Java console application which scrapes a portion of the Sainsbury's Groceries website.

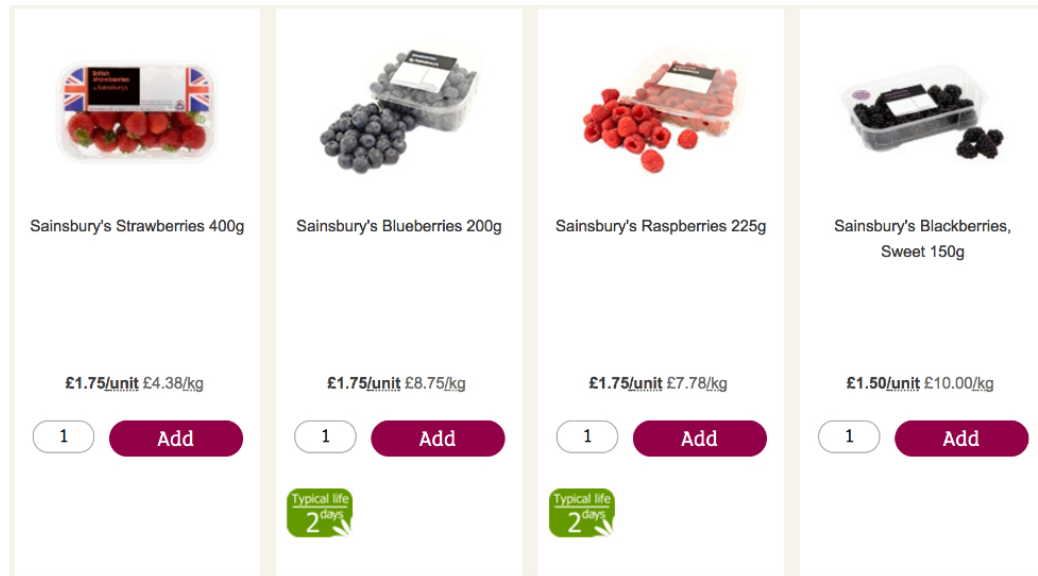
We are looking:

- For clean, concise and testable code.
- At how you have chosen and used tools to help.
- At your app structure as a whole.

Requirements

Using best practice coding methods, build a Java console application that scrapes the Sainsbury's grocery site's - Berries, Cherries, Currants page and returns a JSON array of all the products on the page.

For example:



- Follow each product's link to get the calories per 100g (in kcal) and its long description to include in the JSON.
- Each element in the JSON results array should contain `title`, `unit_price`, `kcal_per_100g` and `description` keys corresponding to items in the HTML.
- Do not include cross sell items, such as the *Sainsbury's Klip Lock Storage Set*.
- Omit the `kcal_per_100g` field, if calories are unavailable.
- If the description is spread over multiple lines, scrape only the first line.
- Show unit price and total up to 2 decimal places, representing pounds and whole pence.
- Additionally, include a total field. This contains two sub-fields: `gross` and `vat`. `gross` will be the total of all the items on the page. the `vat` will be the VAT on the gross amount.
 - For example with a gross of £5.00, the VAT would be £0.83.
 - In this test, every item has 20% VAT.

The link to use is: <https://jsainsburyplc.github.io/serverside-test/site/www.sainsburys.co.uk/webapp/wcs/stores/servlet/gb/groceries/berries-cherries-currants6039.html>

Example JSON:

```
{
  "results": [
    {
      "title": "Sainsbury's Strawberries 400g",
      "kcal_per_100g": 33,
      "unit_price": 1.75,
      "description": "by Sainsbury's strawberries"
    },
    {
      "title": "Sainsbury's Blueberries 200g",
      "kcal_per_100g": 45,
      "unit_price": 1.75,
      "description": "by Sainsbury's blueberries"
    },
    {
      "title": "Sainsbury's Cherry Punnet 200g",
      "kcal_per_100g": 52,
      "unit_price": 1.5,
      "description": "Cherries"
    }
  ]
}
```

```
}  
  
},  
"total": {  
  "gross": 5.00,  
  "vat": 0.83  
}  
}
```

Tools, libraries and frameworks

- Your code should be written in Java 1.7+.
- The choice of tools, libraries and frameworks used to develop the application is left open.
- The application will be tested on a Unix-based terminal and ran in Java 1.8
- Please provide a link to a public VCS, such as Github.

Assessment

We will assess the task based on the following criteria:

- How clean, modular and extensible the code is.
- Suitability of tools, libraries and frameworks used (for both the app itself and any build processes involved).
- Good commit history with meaningful comments.
- Following your instructions gives the correct JSON output.
- Unit or Behavioural Testing, including error handling.
- Anything above and beyond.

Please include a README.md file in the root describing how to run the app, how to run tests and any dependencies needed from the system.

Time

Although it is desirable you complete the test, we appreciate your free time is important. Therefore, if you feel your unfinished code demonstrates the assessment criteria and can be executed, please leave some notes in your README.md explaining what you have written and what you would do next. We would be happy to review this.