Take Home Exercise - Nutrition (.NET)

Format

We'll send you a mostly finished codebase and ask you to work through a few coding tasks in your own time. Allow 1-2 hours to complete the tasks. The code will be sent to you as a ZIP file containing a local Git repository. If you're familiar with Git, please commit your changes to the Git repository (one commit per task), add the repository to a ZIP file with clear instructions on how to run it and then upload your completed solution via the encrypted link in the email the day before your interview (no later than 16:00).

The interview itself will consist of a 30min video call with 1 or 2 xDesign engineers. We'll ask you to walk us through the code changes you've made and we'll provide you with feedback. You can expect some follow-up questions and if time permits we may ask you to make some small code changes.

The Codebase

A simple web service which supports searching and sorting nutrition data. Query results are returned as a list of foods and drinks, formatted as a JSON document.

Querying the data

A number of query parameters are available to customise the search criteria. All query parameters are optional and may be applied in any order. The full list of query parameters is:

- fatRating (Low | Medium | High) - filter by fat rating. When this parameter is omitted, all entries are returned. Categorisation based on `Saturated fats`
- minCalories - Items will match if they are greater than or equal to the minimum calories.
- maxCalories - Items will match if they are less than or equal to the maximum calories.
- limit - Limit the number of items returned. Must be a positive integer.
- sort - Items may be sorted by name or calories or both. The sort parameter is specified as the name of the field, followed by an underscore (_) and then the sort order (asc or desc), e.g. calories_desc. To sort by both name and calories, specify two sort parameters in the query string, one for each field. The order of the fields determines the final order of the items, e.g. if sorting by calories descending and name ascending, the second sort parameter is only used when the calories are equal.

Worksheet

As a minimum, you should try and complete the first two tasks. We recommend completing the remaining tasks in the suggested order but if you're unable to complete the task, you can skip it and move onto another task. If you can't complete one of the tasks, please send us your code changes anyway and we can discuss it further at your interview.

If you're familiar with Git, please commit your code changes for each task in a separate commit. Below are the tasks we'd like you to complete:

1. Implement the filtering correctly. Once you've done this all tests in NutritionControllerFilterTests.cs should pass.

2. Imagine you've been asked to review this code. Considering only the NutritionSearchService class, what comments would you make to the author? Please bring your notes to the interview.

3. The existing solution is hard-coded to load data from a CSV file. We have a new requirement to support loading data from additional file formats, e.g. XML or JSON files.
   a. Refactor the existing codebase to create a simple framework for loading data from other file formats.
   b. The existing CSV parsing should be refactored to work within this framework.
   c. As a part of your refactoring you may create any new classes to support your design.
   d. The choice of file format may be made in configuration or dynamically chosen at startup, but will not change once the application is running.
   e. There's no need to implement the parsing of additional formats but your design should make it clear how these formats will be supported.
   f. At the end of this task, there should be no test regressions and the application should continue to work with the existing CSV data.

4. Fix any bugs(s) causing test failures in NutritionControllerAdditionalTests.cs.

Hints

- You will need .NET 7 to compile the project.
- Tests are split into two files:
  - xDesign.Nutrition.Tests/ControllerTests/
    - NutritionControllerFilterTests.cs should pass on completion of step 1
    - NutritionControllerAdditionalTests.cs are the target of task 4
- The test code and test fixtures (JSON files) contain no errors, and you should not modify the existing tests or the test fixtures as part of your changes.
- There are no errors in the CSV file of nutrition data.
- If you succeed in getting the tests to pass, leave the tests enabled for the remaining tasks. This will help avoid any regressions and the tests can be used to verify your solution to the refactoring.
- The refactoring task is not dependent on fixing the bug.
- Please contact talent.team@xdesign.com if you have any questions and we'll respond within normal business hours (Mon-Fri, 09:00 - 17:00).