Take Home Exercise - Munros (Java)

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 consists of a single RESTful endpoint that gets a sortable, filterable list of Scottish hills (categorised as Munros or Munro Tops), formatted as JSON.

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:

- Filter by category, e.g. 'Munro' or 'Top'. When this parameter is omitted, all hills are returned.
- Filter by minimum height, specified to the nearest tenth of a metre. Hills will match if they are greater than or equal to the minimum height.
- Filter by maximum height, specified to the nearest tenth of a metre. Hills will match if they are less than or equal to the maximum height.
- Limit the number of hills returned. This must be a positive integer.
- Sort the hills by 'name', 'height' or both.

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.

1. Implement the filtering correctly so that all tests in `FilteringTest` pass.
2. Fix any bugs(s) causing test failures in `AllParamsTest`.
3. Imagine you've been asked to review this code. Considering only the `HillSearchService` class, what comments would you make to the author? Please bring your notes to the interview.
4. Refactor the `HillSearchService` class to support loading data from different sources, e.g. XML or JSON files.

Hints

- You will need JDK 17 to compile the project.
- The project is built using Gradle. You can import the project into your favourite IDE or run Gradle from the command line, e.g. `gradlew build`
- `FilteringTest` and `AllParamsTest` are static inner classes of `HillControllerTest`. Remove (or comment out) the `@Disabled` annotation to include the tests in the build.

- The test code is semantically correct and you should not modify the existing tests as part of your changes. There are no errors in the CSV file of hill data.
- If you can't find the bug(s) causing the test failures in `AllParamsTest`, you can disable the tests in `AllParamsTest` and continue with the other tasks. 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).