Dexory – Senior Full-Stack Engineer

Coding Exercise

Overview

This coding exercise is part of the application process for a software engineering role at Dexory. It involves the implementation of a small web application which can perform a comparison between two files and produce a report describing the correlation between the contents of both files.

This is not intended to be a difficult and time-consuming task. It's a way for you to demonstrate the approach you take to writing software – how you follow specifications, how you provide documentation, the way you test functionality, and how you make decisions about what to build. It also gives us an opportunity to discuss this process with you afterwards.

This task is closely based on a real-world problem we have previously had to solve for a customer application, and is broadly representative of the kind of tasks you would perform in this role.The specification is fairly minimal and you are expected to resolve in a reasonable way given the information available.

The task

Our robot systems scan a customer warehouse automatically every evening and produce a report describing the location of assets. This information is made available as a JSON file, which a robot automatically uploads to an HTTP endpoint at the end of a scan. This JSON file consists of a list of all locations that were scanned in the warehouse, and indicates if they were occupied during the scan, along with any barcodes that were scanned in the location.

Separately, the customer will upload a CSV file containing a list of the items that they expect to have been located in each location in the warehouse during the scan.

The requirement is to provide an application which can compare the data gathered by the robot and the data supplied by the customer, and generate a report that identifies any discrepancies. The output should provide a result for comparing each location, and consist of:

- The name of the location
- Whether or not the location was successfully scanned
- Whether or not the location was occupied
- The barcodes that were expected to be found in this location
- The barcodes that were actually found in this location

- A description of the outcome of the comparison, using at least the following statuses:
    - The location was empty, as expected
    - The location was empty, but it should have been occupied
    - The location was occupied by the expected items
    - The location was occupied by the wrong items
    - The location was occupied by an item, but should have been empty
    - The location was occupied, but no barcode could be identified

Functional requirements

- A web application is available which can consume JSON from a robot and store it in an appropriate data store.
- A user can select a previously-uploaded JSON payload, and upload a CSV report to generate a comparison report.
- A user can view or export the comparison report in an appropriate format.
- The JSON and CSV upload endpoints accept the format supplied in the sample files accompanying this coding exercise.

Non-functional requirements

- You should include appropriate documentation about your code.
- Your code should follow best practices and a consistent coding style.
- Your submission should include whatever test cases you consider appropriate.

Data and deliverables

Included with this document are the following files:

- **`example-robot.json`** – an example JSON payload from a robot.
- **`example-customer.csv`** – an example CSV payload from a customer.

You should deliver:

- Your code that implements the process described above.
- Any documentation or test cases you have produced.
- Details about how the code can be built and executed.
- An example of the comparison report generated using the supplied example files.

You can submit your results in a Git repository or a ZIP file. We encourage you to reach out to us with any questions you may have about the task before submitting your response.

Good luck and have fun!