Data Engineer Task

Overview

The candidate is required to transform a provided MongoDB collection of restaurant brands, containing various inconsistencies and errors, into a standardized format using a provided Mongoose schema. Additionally, the candidate must extend the dataset with new seed data.

Task Details

1 - Data Transformation

You will be provided with a MongoDB collection named brands (Brands collection: brands.json), containing 10 documents representing sample brands. These documents have intentional mistakes in their schema, such as incorrect field names, types, and validations.

Your task is to write a TypeScript code project using Mongoose to transform this data into a correct format based on a given schema: Brands schema: brands-schema.ts

Transform the data in-place in the same documents (same object Id) and the same collection. Import the file first to a MongoDB database then apply the transformations on the brands collection in your database. Do not migrate or re-save the data to another database during transformation.

Ensure that the data is validated against the schema during the transformation process.

Notes

- If the yearFounded or numberOfLocations field is available in another field with a different name in the same document, you should get it from that other field.
- If the yearFounded or numberOfLocations field is not available at all in the document with a correct data format, use the minimum year as per the schema.
- Your code should be specific only for the Brands schema (not to any generic schema), so you can use specific field names in your code, etc.

2 - Data Seeding

- Extend the database by generating 10 new brand documents with correct schema adherence.
- Use any data library (e.g., Faker.js) to create test data for the new entries with different cases.
- Document the seed data cases in an Excel file to explain what differentiates each case.

3 - Export the Brands collection

- After you transform the data within the same database and after seeding the test data, export the brands collection as a json file.

Technologies and Frameworks

- Please ensure you use Node.js in TypeScript, with Mongoose library.
- Feel free to use Nest.js if preferred.

Evaluation Criteria

- Accuracy of data transformation and adherence to the provided schema.
- Logical and efficient approach to identifying and correcting data inconsistencies.
- Quality and readability of the TypeScript code.
- Completeness and clarity of the process documentation.

Code and Data Submission

- Push your TypeScript code, along with any support files, to a public GitHub repository under your GitHub account.
- Include the modified brands.json file (exported from your MongoDB database) in your git repository.
- Attach the Excel documentation file to your git repository.
- Respond to the assessment task email with the links of the github repositories.

Have any questions ?

For any questions or inquiries feel free to contact us at careers@pleny.com

Good luck, and we look forward to your innovative solutions!