Menu Management Backend

Instructions

- Use Rails or another web app framework you're familiar with.

- Place in a Github Repo and send us a link when you're done.

- The project is due 1 week after receiving this document, but you can work on it as much as you'd like.

- Write extensible code, make commits, and test as you would a production application. We'd rather see a well-constructed and well-tested app with fewer features.

- Complete as many levels as you wish, but complete them one at a time. Resist making decisions for future levels. Your commits should reflect this approach.

- Do not hesitate to ask us any questions you've got.

Requirements

- Level 1: Basics

  - Create an object model for `Menu` and `MenuItem` s classes.

  - `Menu` has many `MenuItem` s.

  - `Menu` and `MenuItem` have typical data associated with restaurants.

  - Illustrate behavior via unit tests.

- Level 2: Multiple Menus

  - Introduce a `Restaurant` model, and allow `Restaurant` s to have multiple `Menu` s.

  - `MenuItem` names should not be duplicated in the database.

  - `MenuItem` can be on multiple `Menu` s of a `Restaurant` .

  - Illustrate behavior via unit tests.

- Level 3:
  - Convert the following json body from the included file, `restaurant_data.json` in to our new menu system.
  - Make any model/validation changes that you feel are necessary to make the import as complete as possible.
  - The output should return a list of logs for each menu item and a success/fail result.
  - Apply what you consider to be adequate logging and exception handling.
  - Illustrate behavior via unit tests.

Evaluation

To ensure consistency, our Engineering team uses a rubric to score your submission according to the following criteria. Please show us your strengths for coding, testing, technical design, and attention to detail.

Changeability:

- We expect an extensible project. If requirements change, your code should be adaptable.

Iterative approach:

- We expect an iterative approach to complete each level. We appreciate it when you make us aware of the assumptions you used and decisions you made along the way.

Verification:

- We expect a low-bug or bug-free implementation of the project.

Validation with tests:

- We expect application behavior to be validated with unit tests.