

NodeJS - Assessment

Introduction

This week you will be building APIs in the node server for Food Order Application.

We have provided the starter code for the Node Server which you will be working on.

To run the Node Server App in the localhost kindly follow the below mentioned steps -

1. Download and extract the **Food_Order_App_Node Server** folder from the **NodeJS_Assessment_For Noncoders_Source Code.zip** folder.
2. Open a terminal and navigate to the **Food_Order_App_Node Server** folder in it.
3. Run the `"node server.js"` command to run the application in the localhost. This will run your application on localhost on port 8080.
4. Now open the Postman tool to connect to the APIs of the node server. The base URL for the node server is <http://localhost:8080/v1/>
5. You can try connecting to APIs as per the endpoints provided. For ex, if you want to fetch all categories you will give GET URL as <http://localhost:8080/v1/categories> and if you want to fetch specific category you will give its id in the query parameter for eg- <http://localhost:8080/v1/categories?id=10> will fetch category with category Id 10.

Housekeeping points

- This is a minimal example and may not follow some standard practices.
- We focus on the main flow, and not much error handling.

Problem Statement

Your task is as follows -

1. **Implementation of APIs to fetch cuisine details (URL Example - <http://localhost:8080/v1/cuisines>, <http://localhost:8080/v1/cuisines?id=10>)**
 - a. **Cuisine Handler** - Complete the code of `fetchAllCuisinesHandler()` and `fetchCuisineByIdHandler()` to check the request method and accordingly invoke different APIs from `cuisineService.js`
 - b. **Cuisine Service** - Complete the code of `cuisineService.js` to define 3 APIs - to fetch all cuisines, to fetch a cuisine by Id and to fetch a cuisine by name.
2. **Implementation of APIs to fetch restaurant details (URL Example - <http://127.0.0.1:8080/v1/restaurants>, <http://127.0.0.1:8080/v1/restaurants?id=1>)**
 - a. **Restaurant Handler** - Complete the code of `fetchAllRestaurantsHandler()` and `fetchRestaurantByIdHandler()` to check the request method and accordingly invoke different APIs from `restaurantService.js`
 - b. **Restaurant Service** - Complete the code of `restaurantService.js` to define 2 APIs - to fetch all restaurants and to fetch a restaurant by Id.

Program Organization

- You will be getting a folder named **NodeJS_Assessment_For Noncoders** having a sub-folder named **Food_Order_App_Node Server** which further has a subfolder named **app** and a file containing server.js file.
- All the data files are present in the data folder of the app folder.
- All the handler files are present in the handler folder of the app folder.
- All the service files are present in the service folder of the app folder.

Evaluation Rubric

Total Project Points: 60

- Correctness:
 - Correctness of implementation
 - Problem statement - point 1 : 30 Points
 - Problem statement - point 2 : 30 Points

Program Instructions

- Make sure you zip the **NodeJS_Assessment_For Noncoders** folder before submitting the project.
- Project will not be evaluated if the submitted project is not in the zip/rar format.