Processing Map Data - Project Documentation

Author: Aashish Kulkarni

Date: 31/03/25

---

1. Introduction

1.1 Project Overview

This project focuses on processing location and metadata JSON files to extract useful insights such as:

- Counting valid locations per type (e.g., restaurants, hotels, cafes, etc.)
- Calculating the average rating per type
- Identifying the most reviewed location
- Detecting locations with incomplete data
- Displaying results in a structured HTML table format

The project uses **Spring Boot** for backend processing and **JavaScript (AJAX)** for fetching and displaying the data dynamically.

2. Approach

2.1 Loading Data

The application loads two JSON files:

- location.json - Contains location details (ID, latitude, longitude)
- metadata.json - Contains metadata details (ID, type, rating, reviews)

Spring Boot reads these files and merges the data based on the `ID` field.

2.2 Data Merging

- The backend service (DataService.java) iterates through both JSON files.
- It matches locations and metadata by `ID`.
- The merged data is structured as a list of maps.
- If an entry lacks metadata (type, rating, or reviews), it is flagged as incomplete.

3.Backend – Java (Spring Boot)

The backend is built using **Spring Boot**, which loads and processes location and metadata JSON files. It provides RESTful API endpoints to merge data, calculate statistics, and identify incomplete entries. The application ensures efficient data handling and processing using Java collections and mapping techniques.

4. Frontend - HTML & JavaScript

The frontend consists of a simple **HTML** structure with interactive buttons to fetch data dynamically using **JavaScript (AJAX)**. It uses **jQuery** to make API calls and updates the results in structured **tables**, providing an intuitive user experience.

5. Challenges & Solutions

5.1 Handling Missing Data

- **Problem:** Some metadata fields (type, rating, reviews) were missing, causing errors.
- **Solution:** Used null checks and assigned default values like `" Missing"`.

5.2 Merging JSON Data

- **Problem:** Some locations lacked corresponding metadata, leading to mismatched IDs.
- **Solution:** Implemented a mapping approach that ensures all locations are included, even if metadata is missing, and flagged incomplete entries.

5.3 JavaScript Not Executing in External File

- **Problem:** Moving JavaScript to an external file (script.js) initially failed.
- **Solution:** Corrected file path and ensured it was loaded after jQuery.

6. Future Enhancements

- Improve UI with filters and sorting.
- Optimize backend processing for large datasets.
- Implement database storage instead of static JSON files.

7. Conclusion

This project successfully processes location data, performs analytical operations, and displays results in an interactive UI. The modular design allows for future scalability and enhancements.