

## app\_js

### Explanation:

This JavaScript code builds an interactive dashboard to visualize data from a JSON file, using D3.js and Plotly.js. It adheres to a provided framework and prioritizes efficient data handling, clear display, and robust error management.

#### 1. buildMetadata(sample):

- \* **Data Fetching and Error Handling:** Fetches data using `d3.json()`, includes a `.catch()` block for robust error handling.
- \* **Metadata Display:** Clears previous metadata using `metadataPanel.html("")` to prevent duplication. Displays key-value pairs in the `#sample-metadata` element.

#### 2. buildCharts(sample):

- \* **Data Fetching and Error Handling:** Uses `d3.json()` with a `.catch()` block for error handling.
- \* **Bubble Chart:**
  - \* **Visualization:** Uses `otu_ids`, `sample_values`, and `otu_labels`. Applies a colorscale (e.g., 'Earth') for enhanced visualization.
  - \* **Layout:** Adds an x-axis title ("OTU ID") for clarity. Renders in the "bubble" HTML element.
- \* **Bar Chart:**
  - \* **Labels:** Formats OTU IDs as OTU `{otuID}` for clear labeling.
  - \* **Data Handling:** Uses the top 10 `otu_ids`, `sample_values` (reversed), and `otu_labels` (reversed). Displays horizontally in the "bar" HTML element.

#### 3. init():

- \* **Data Handling:** Efficiently fetches data once with `d3.json()` and includes error handling.
- \* **Dropdown Population:** Populates the `#selDataset` dropdown with sample IDs.
- \* **Initial Display:** Calls `optionChanged` initially to display the first sample's data immediately, enhancing the user experience.
- \* **Data Integrity:** Ensures correct filtering and display logic for the initial data.

#### 4. optionChanged(newSample):

- \* **Interactivity:** Updates the dashboard when a new sample is selected.
- \* **Data Handling:** Calls `buildCharts` and `buildMetadata` with the new sample ID.
- \* **Error Handling:** Includes a `.catch()` block for robust error handling during data processing, logging errors to the console and displaying simple, customizable error messages on the page.

**Key Features:**

- **Error Handling:** Comprehensive `.catch()` blocks for robust error management throughout the code.
- **Clarity and Readability:** Code is formatted for easy readability.
- **Efficiency:** Data is fetched once in `init()` for optimal performance.