# Code Inspection for historical-data.js

**Description:**

The historical-data.js file powers the interactivity of historical.html, managing server-side data fetching and visualization for a DataTable and Chart.js graphs. It handles table initialization, graph plotting, and filter updates, ensuring a dynamic, responsive user experience with robust error handling and debugging logs.

**Functionality Description:**

* **Setup:** 
  1. Triggers on DOMContentLoaded, initializes table and graph with 500ms delay.

○ API base: http://localhost:5000/api/historical/.

* **Data Structures:** 
  1. availableColumns: Defines line-specific columns (e.g., line4: timestamp, r01-r08; line5: timestamp, timezone, r01-r17).

○ currentGraphData: Stores fetched graph data for reuse.

* **Interactive Elements:** 
  1. **Tabs:**

■ #table-tab: Shows table, hides graph.

■ #graph-tab: Shows graph, fetches or replots data.

○ **Filters:**

■ #lineSelector: Switches lines, updates table/graph.

■ #dateFilter: Refreshes data on change.

■ #searchBox: Updates on keyup.

○ **Chart Type (#chartTypeSelector):** Changes graph type (line, pie, bar, area, scatter).

○ **Table (#lineTable):** Server-side DataTable, dynamic columns, timestamp sort.

○ **Graph (#dataChart):** Renders data with Chart.js, five chart types.

* **Data Fetching and Processing:** 
  1. **Table:** POSTs to API with filters, reloads on change.

○ **Graph:** Fetches once, stores data, replots on type change, uses random colors.

**Links:**

* **API:** http://localhost:5000/api/historical/{line}.
* **Libraries:** jQuery, DataTables, Chart.js (loaded in historical.html).

**Extra Details – Interactive Elements:**

* **Event Listeners:** Tab clicks, filter changes, chart type updates.
* **Table Features:** Server-side, responsive, descending timestamp sort.
* **Graph Features:** 
  1. Line: Multi-sensor, timestamp x-axis.

○ Pie/Bar: Sensor averages.

○ Area: Filled lines.

○ Scatter: Timestamp-sensor points.

* **Error Handling:** Console logs, alerts for API/data issues.

**JS Explanation and Application:**

* **Initialization:** Ensures DOM readiness with delay.
* **Table:** Dynamic, AJAX-driven, filter-responsive.

**Graph:** Efficient data reuse, flexible chart options.

● **Interactivity:** Seamless updates, robust logging.