Frontend JavaScript Function Reference

This document catalogs the key JavaScript functions found in the frontend files (app.js for Cleanup and management.js for Management), detailing their role in handling user interaction and API communication.

1. Utility Function (Shared)

This function is a simple helper used across both applications for consistent visual feedback.

displayMessage(type, message)

Detail	Description
Purpose	Displays a transient success or error message to the user in the designated message area (#messageDisplay).
Parameters	type (string: 'success', 'error', or 'message'); message (string: The text content).
Called By	All primary action functions (submitMapping, batchCreateAndMap, saveChanges).

2. Company Cleanup Interface (app.js / cleanup.html)

These functions manage the raw name list, suggestion search, and the three mapping actions (map existing, create new, skip).

loadUnmappedList()

Detail	Description
Purpose	Fetches the full list of unmapped raw company names from the backend and populates the sidebar.
API Call	GET /api/unmapped_list

Called By	**Initial Load (in cleanup.html):** document.addEventListener('DOMContentLoaded', loadUnmappedList);
Called By	**Post-Action:** Called after a successful mapping/skip/batch action to refresh the list.

selectRawName(name, element)

Detail	Description
Purpose	Updates the detail panel to show the selected raw name and enables the mapping actions.
Parameters	name (string: The raw company name); element (DOM element: The list item clicked).
Called By	**HTML Context (cleanup.html List):** Dynamically attached onclick event to each list item generated by loadUnmappedList.

fetchSuggestions(inputId, datalistId)

Detail	Description
Purpose	Queries the backend for clean company names to provide autocomplete suggestions to the user.
API Call	GET /api/suggestions?q={search_term}
Called By	**HTML Context (cleanup.html Input):** Event listener attached in app.js to the search input element.

submitMapping(mappingType)

Detail	Description
Purpose	Sends the user's mapping decision (map, create, or skip) to the backend.
API Call	POST /api/map_company
Parameters	mappingType (string: 'map', 'create', or 'skip').
Called By	**HTML Integration (cleanup.html):**
	Existing Map: <button onclick="submitMapping('map')">
	New Create: <button onclick="submitMapping('create')">
	Skip: <button onclick="submitMapping('skip')">

batchCreateAndMap()

Detail	Description
Purpose	Processes all remaining unmapped names in a batch, creating a new, self-mapped company profile for each, and setting target_interest: true.
API Call	Repeated POST /api/map_company (once per raw name).
Called By	**HTML Integration (cleanup.html):** Attached directly to the batch button in the HTML.

3. Company Management Interface (management.js / management.html)

These functions handle loading the clean company profiles, viewing details, and saving edits.

loadCompanyList()

Detail	Description
Purpose	Fetches the list of all clean company profiles and populates the management sidebar.
API Call	GET /api/companies
Called By	**Initial Load (in management.html):** document.addEventListener('DOMContentLoaded', loadCompanyList);
Called By	**Post-Save:** Called by saveChanges() to refresh the sidebar name.

selectCompany(id, element)

Detail	Description
Purpose	Handles a click on a company list item, highlights the item, and triggers the detail load.
Parameters	id (integer: The company's company_id); element (DOM element: The list item clicked).
Called By	**HTML Context (management.html List):** Dynamically attached onclick event to each list item generated by loadCompanyList.

loadCompanyDetails(companyId)

Detail	Description
Purpose	Fetches the complete profile data for a single company and populates the edit

	form fields.
API Call	GET /api/companies/{company_id}
Called By	Called internally by selectCompany(id, element).

saveChanges()

Detail	**Description**
Purpose	Collects all data from the detail form fields and sends a PUT request to update the profile in the database.
API Call	PUT /api/companies/{company_id}
Called By	**HTML Integration (management.html):** Attached directly to the Save button in the HTML form.