Functional Specifications

New Feature Summary
This document contains functional specifications for sprint 0 of the project to rewrite the Evergreen acquisitions client module in Angular framework. Sprint 0 is concerned with infrastructure and will lay the groundwork for future sprints.

Overview
Five pieces of infrastructure will be created during this sprint:
1. Adoption of a JavaScript library to accurately perform monetary calculations in the client
2. Enhancement of the eg-grid widget to:
   a. Add per-column results filtering
   b. Functionality to ensure that when editing a single record in a grid using a modal, that saving the change to the record will result in a refresh of the record in question without losing the user’s place in the grid
3. Enhancements to the eg-combobox widget to:
   a. Reduce boilerplate code needed to retrieve results from a single known IDL class
   b. Allow the user to use the drop-down button to retrieve an initial set of results from a dynamic source
4. Implementing support for cross-tab communications
5. Adding a new help popover widget.

Current behavior
The Angular client introduced in Evergreen 3.2.0 already includes eg-grid and eg-combobox widgets.

Development Plan
User interface
Staff user interface details
eg-grid
The eg-grid widget will gain options to implement generic result filtering similar to the functionality of openils.widget.PCrudFilterPane. We propose two options for presenting this functionality.
**eg-grid Option A: Display filter controls inline**

With this option, the filter controls will be displayed inline, below the grid column headings. This feature will be controlled by several attributes, including:

- *(eg-grid)* enable-result-filter: Boolean (default value false) to control whether to offer result filtering for a grid instance
- *(eg-grid-column)* exclude-from-result-filter: Boolean (default value false) to specify that a column should not be an option for the result filter
- *(eg-grid-column)* result-filter-underlying-field: This is an optional attribute to specify an IDL path of the field to use to filter values by, and would be used if the value displayed in the grid for that column is derived from multiple fields.

If *enable-result-filter* is on, the grid will display, below the column headings, a row of input widgets. The user will be able to interact with those widgets to specify a value and optional conditions to filter the results by. Upon entering a value in a filter column, the grid results will be retrieved with the appropriate conditions in place.

The type of the filter input widget be based on the IDL type of the underlying IDL field for that column:
- text: text input next to a drop-down allowing selection of an operator
  - is
  - is not
  - is null
  - is not null
  - is less than
  - is greater than
  - is less than or equal to
  - is greater than or equal to
  - is like
  - is not like
- numeric: numeric input next to a drop-down allowing selection of an operator
- date: calendar selection widget next to a drop-down allowing selection of an operator

The user will be able to clear the value for a given filter. The user will also have an option to clear the value of all filters. Column headers will be able be frozen at the top of the grid and filtering will be applied across the full dataset, not just the visible rows.

**eg-grid Option B: Display filter controls via a separate modal**

In this scenario, the filter controls would be displayed either in a modal (similar to the Dojo interface for grid filter) or in a div displayed above the grid column header.

This feature will be controlled by several attributes, including:

- *(eg-grid)* enable-result-filter: Boolean (default value false) to control whether to offer result filtering for a grid instance
- *(eg-grid)* use-modal-for-result-filter: Boolean (default value true) to control whether the result filter controls display in a modal or inline just above the grid column header
- *(eg-grid-column)* exclude-from-result-filter: Boolean (default value false) to specify that a column should not be an option for the result filter
If enable-result-filter is on, the grid will display a button or text link labeled "Filter" or similar. Clicking on that link will display an interface that would allow the user to select one or more columns from the grid, an operation (e.g., is, is not, is not null, is greater than, etc.), and a value. Upon choosing to apply the filter, grid results would be retrieved with the appropriate conditions in place. If more than one filter column is specified, an implicit “AND” will be applied.

Other changes to eg-grid

eg-grid will also gain an option to keep track of the user’s current place when paging through grid rows even after a refresh. This will become the default option, although developers will have the ability to specify that a grid refresh intentionally “forgets” a user’s place if as that may be warranted in specific contexts.

Additionally, eg-grid will highlight the last selected row in the grid in order to provide a visual cue to the user.

eg-combobox

The existing eg-combobox widget will be used as the basis of a new eg-fm-combobox widget that will have the following attributes:

1. Passed to eg-combobox
   a. name
   b. placeholder
   c. persistKey
   d. allowFreeText
   e. required
2. idlClass: IDL class of that the eg-fm-combobox is meant to retrieve values from
3. leftAnchored: Boolean (default false) to indicate whether searches done during type-ahead retrieval should be left-anchored
4. fetchOnDownArrow: Boolean (default false) to indicate whether to fetch an initial set of results if the user clicks on the button of the combobox without having entered any text.

eg-help-popover

A new widget similar to the AngularJS egHelpPopover widget will be created to allow the insertion of additional help text. eg-help-popover will display a question mark or similar logo; when clicked or selected, additional text will display.

The popover text will display at an optimal location on the screen regardless of where the help widget is positioned.

The Angular user interface sandbox (e.g., "/eg2/en-US/staff/sandbox") will be updated to include working examples of eg-grid result filtering, eg-fm-combobox, and eg-help-popover.

Workflow changes

This sprint does not change existing workflows.

Settings and Permissions

This sprint does not add new settings or permissions.
Implementation Areas
As this sprint is concerned with UI components for the Angular staff interface, it will make no changes to databases, indexing, core business logic, or seed data, and will not in itself result in any upgrade deviations or require a reingest.

Angular component changes
In addition to the UI changes discussed above, a new Angular service, broadcast, will be added that will use BroadcastChannel and window.postMessage to permit messages to be sent between related browser tabs. This service will have the following characteristics:
- the ability to register a routine to be invoked upon receiving a message from a known BroadcastChannel
- a wrapper around window.open and window.postMessage to permit bi-directional communication between a tab and the tab that opened it

Security Consideration Planning
Work in this sprint raises no additional security considerations.

Quality Assurance Planning
Performance Implications or Concerns
There are no particular performance concerns associated with this work.

Recommended Unit and Regression Testing
Angular unit tests will be written for the new and updated components.

Recommended Compliance and Accessibility Testing
We will develop new features in accordance with existing OPAC WCAG compliance, and will test usability in desktop versions of Chrome and Firefox.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. Please see the end of this document for the terms of the license.
Mockups

Mockups are for illustrative purposes only and may change during the implementation due to developer discretion.

Option A: Display filter controls inline
Option B: Display filter controls via a separate modal
Tooltips

Select Dry Run to see a preview of the changes that would occur based on the selected actions. Data will not be changed when Dry Run is selected.
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Here is an example of a standard attribution:
"PROJECT NAME & HYPERLINK TO EOLI POSTING" by Equinox Open Library Initiative is licensed under CC BY-SA 4.0

Please be aware that under the CC BY-SA 4.0, if you share the licensed material (including in modified form), you must:

- retain the following if it is supplied by the Licensor with the Licensed Material:
  - identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor
  - a copyright notice;
  - a notice that refers to this Public License;
  - a notice that refers to the disclaimer of warranties;
  - a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
  - indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
  - indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.