# G3Tools on DataJalapeno Platform System Design

## Table

* List of items is retrieved from database and put in Vuex store when list is created. If we route to another page and come back, the list should not refresh.
* List of items is maintained in store and is automatically refreshed every 10 minutes.
* User can manually refresh list at any time by clicking on the refresh action button on the list.

### Add Button

When on a list showing resources (like brands), an Add button allows the user to create a new brand record.

Main goals:

* Sub-second response adding a record
* Optimistic database interaction: We will assume the record will save and so we can show the new record immediately in the list and save the record in the background. If an error occurs, we back out the record from the UI and show the error.

Process:

1. ✓User clicks on Add button from table action bar (top right)
2. ✓Item detail for this resource displayed (via router), passing mode=add as props. (Note, the same item detail component can be used for add, change, or view purposes, depending on mode passed in.)
3. Validation:
   * Save button is disabled until fields are changed.
   * Required fields and custom validations provide immediate feedback if input is in error.
4. ✓User fills in data into fields.
5. User clicks save.
   * Audit fields *created on*, *created by*, *modified on*, and *modified by* are added to the record by the UI.
     + The database only adds created and modified on/by as a fallback, in case the UI doesn’t add it.
   * A GUID (temporary unique id) is added to the new item record by UI.
   * User is brought back to list immediately with new item appearing at the top of the list (regardless of sort).
   * Toast is displayed immediately, such as, “ ‘New Brand’ added to database” with Undo button displayed, and remains for 7 seconds.
   * If user *doesn’t* click undo, toast expires after 7 seconds and record is saved.
     + Record is added to database with a put call.
     + If an error occurs and the record is not saved, remove the record from the list and display the error to the user.
     + If save is successful, new saved record is returned from API. If still on the list, the list record is replaced by the new returned record (with the database-assigned id), or just id is replaced.
     + Toast disappears.
   * If within 7 seconds, the user clicks undo,
     + New record is removed from table list
     + Toast closes and nothing is added to database
6. User clicks cancel:
   * If data has been entered into item detail scren, confirm if we should save before canceling.
   * If Yes, save (see above)
   * ✓If No, discard. Item detail closes and user is brought back to resource list.
   * List page should not be refreshed from the database. (Vuex?). When we come back to the list from the detail page through cancel or back, don’t refresh. But, when we route to the list from somewhere else? Maybe refresh.