

KRAD - Collections UI - Add via editable blank line, no additional Add needed

Purpose

This is part of a set of UI requirements related to the collections UI, specifically, in adding collection items.

In the current design, many users tend to enter data into the fields in the visible add line but then don't select the add button, so the data doesn't get added, and can be lost on submitting the page. This relates to the current design where the add row is the first in the collection, with the only difference being the add button instead of the delete button (same placement). These users don't realize or notice that activating the add button is required.

Solution: Add via editable blank line. See mocks and diagrams section below.

Note 1: There is another UI requirement in 2.2 milestone 2 that creates another way to solve this same problem, via a lightbox. Both of these options should be available to applications.

Note 2: In Rice 2.0, KRAD added the option of moving the add line visually outside of the current collection content and shading it, so that users could more easily perceive its difference from the current collection. That option will continue to be supported in 2.2. Applications will have 3 choices for presenting the Add operation to users.

Detailed Description

See the Mocks and Diagrams section below.

Usage Scenarios

See the Mocks and Diagrams section that follows.

Mocks and Diagrams

There are two variations of this design, described below. They have the same starting point. Note: This applies to table and stacked collections and sub-collections, though most of the examples below show table collections.

Options 1 and 2:

The Add button is placed above the current content of the collection:

Add Line				
* Field 1	* Field 2	* Field 3	* Field 4	Actions
1	2	3	4	delete
A	B	C	D	delete
a	b	c	d	delete
5	6	7	8	delete

Options 1 & 2: When the user activates the Add button, a blank line is added to the collection. The focus is placed in the first field. The blank fields are already saved (without validation), and blanks will be submitted if the user doesn't enter data. The KRAD validation framework will handle in typical fashion - e.g., client-side validation message if the field is required and the user moves focus out of the field, server-side validation message if the field is required and the form is submitted with it blank. See mock-up below (minus the arrows showing the results of activating the Add Line button):

* Field 1	* Field 2	* Field 3	* Field 4	Actions
				delete
1	2	3	4	delete
A	B	C	D	delete
a	b	c	d	delete
5	6	7	8	delete

Option 1:

The row is shaded until it is saved (or until it is submitted, if a user doesn't save before submitting). The delete button is active at all times on this new row, since the blanks have already been added.

Option 2:

In addition to the row being added, with focus in first field, and the row being shaded, an additional 'Save' button is added to the Actions column - see mock up below:

* Field 1	* Field 2	* Field 3	* Field 4	Actions
				delete Save
1	2	3	4	delete Save
A	B	C	D	delete Save
a	b	c	d	delete Save
5	6	7	8	delete Save

Once the user has filled in all the required fields, the Save button becomes selectable:

This is somewhat similar to the current design in that the "Save" button replaces the "Add" button. But it is different in that each new row/collection item is already temporarily "added" (it will be included in a submit), and this "Save" button makes it possible to immediately save and perform additional server-side validation on just that row (collection item). (Note: technical discussion is underway to determine if this is feasible or if the entire collection must be saved/refreshed each time.)

Once a row has been saved, the yellow highlighting is removed and the Save button is greyed out.

If the user makes a change in a field of any row that has already been saved, the Save button becomes active again. See the example mock up below (Note, for M3 we may consider a dirty field indication on the changed fields):

Add Line

* Field 1	* Field 2	* Field 3	* Field 4	Actions	
W	X	Y	Z	delete	Save
1	2	3	4	delete	Save
A	B	C	RF	delete	Save
a	b	c	d	delete	Save
5	6	7	8	delete	Save

Save Check Form Submit Cancel

Variation on Add button and added line placement

The above mockups show the default placement of the Add button above the collection, with the 'just added' item showing up at the top of the collection. In addition to this, KRAD should support the alternative of placing the Add button at the bottom of a collection, with the 'just added' item showing up at the bottom of the collection.

An example of this in a table collection is shown in the mockup below:

Contact Name	Relationship	Phone Number	Email Address	Primary?	Actions
	Select One			<input type="checkbox"/>	delete

Add another Contact

Contact Name	Relationship	Phone Number	Email Address	Primary?	Actions
Dan Smith	Select One	888-555-1212	dansmith@kuali.org	<input checked="" type="checkbox"/>	delete

Add another Contact

Contact Name	Relationship	Phone Number	Email Address	Primary?	Actions
Dan Smith	Spouse	888-555-1212	dansmith@kuali.org	<input checked="" type="checkbox"/>	delete
	Select One			<input type="checkbox"/>	delete

Add another Contact

An example of this placement in a stacked collection is shown in the example below:

Field Label
Don't see your selection? **Contact** an administrator for help.

Advanced Search

Advanced Search

Performance

If applicable, list expectations for performance (optimal and worst cases would be fine, give time in seconds):

References

See earlier work at <https://wiki.kuali.org/pages/viewpage.action?pageId=312351977>

- [KULRICE-5176](#)
- [KULRICE-5570](#)
- [KULRICE-5345](#)
- [KULRICE-5365](#)
- [KULRICE-5367](#)
- [KULRICE-5370](#)
- [KULRICE-5284](#)
- [KULRICE-189](#)

Requirements Listing

List all requirements (individual verifiable statements) that indicate whether the work for this item has been complete. If there are requirements that are not essential to the functionality but would be nice to have if time allows, enter those under 'Secondary':

Primary:

1. Populate both a "Delete" and a "Save" button on the new line when added.
2. Shade the new line yellow until user saves it (use background color #FFFFCC for the shading value on the new items until saved).
3. Remove the yellow shading from lines that are saved (that are moved off the new line list - an internal architectural construct).
4. Populate both a "Delete" and a "Save" button on existing lines that are editable.
5. Disable the "Save" button on these existing lines, by default.
6. Change the "Save" button's state (which by default is greyed out on all lines that are editable but that have already been saved in the data model) to active on any editable line when a user makes a change in a field in the editable line
7. Assumption: KRAD framework can handle the save action at the document level but not at the line level. KRAD will know the line passes all validations and can assume the line save was successful when it does, but can't know that for sure (the application will have to hook in and manage the save appropriately). *Technical Note: Can we create a flag that the application can populate to let us know the save is successful?*

Secondary:

1. Handle the Save at the line level, in the same way we currently do at the document level (handled by the framework, without requiring as much application handling as the above primary reqs list includes).

Dependencies

List any functional or technical work that must be completed before work on this item can begin:

1. item

Issues

List any issues that need to be resolved before work on this item can begin:

Functional:

1. item
2. item

Technical:

1. item
2. item

QA or Regression Testing Plan

List steps needed to test the basic functionality of this update, enhancement, bug fix

1. test/steps
2. test/steps

Checkoff

Functional Analysis Complete? **No (completed by SME)**

Needs Review by KAI? **No (completed by SME)**

Technical Analysis Complete? **No (completed by DM)**

Needs Review by KTI? **No (completed by DM)**

Estimate: **30 hours (completed by DM)**

Technical Design: [Link Here](#) **(completed by DM)**

Jira: [KULRICE-7393](#)

Final Documentation: [Link Here](#) **(completed by DM)**

Added to QA: **No (completed by SME)**