



# Nuxeo

---

Nuxeo Web UI

Drag and Drop Specifications

Last update: 12 / 07 / 2017

# Table of Content

<b>Feature Overview</b>	<b>3</b>
Goals	3
Audience	3
Scope	3
Reference Documents	3
Glossary	3
<b>Functional Requirements</b>	<b>4</b>
User Stories	4
Acceptance Criteria	4
Tests Cases	4
<b>Non Functional Requirements/ Software Attributes</b>	<b>5</b>
Scalability	5
Privacy	5
Performance	5
<b>Features to Implement</b>	<b>6</b>
Requirements	6
Feature(s) description	6
Mockups	6
<b>Limitations</b>	<b>7</b>

# Feature Overview

## Goals

The goal of this topic is to introduced Drag and Drop functionality to improve asset location management and collection-based workflows.

The user should be able to

- **move** items **from containers to containers**
- **add assets to from container to collections**
- **move** or **add assets from search result to collections or containers**

## Problems to solve

Currently, changing asset location is a complex user operation. User selects assets, copies to a clipboard, navigates to targeted container, opens clipboard and moves (a button) documents there. This is not natural for all users and has too many steps in it, even just to move assets to a sibling container or an ancestor.

Another identified problem is user stories that are based on collection usage. Collections can be used to categorize and hold assets in a semantic / bias fashion. Users can share these collections and collaborate together with entities.

But there can be several steps involved to add to a collection. User selects assets, taps on “add to collection”, search a collection or creates a new one.

Side topic from this change:

- if the user can move or copy asset by selection + toolbar or by drag and drop, **the clipboard doesn't make sense anymore and should be removed.**
- If the user want to drop items in the tree, **the tree behavior need to be updated so that you can fold/unfold parents and children without navigates to them.**

## Audience

Product, engineer and QA teams.

## Scope

Drag and Drop asset functionality and small collection / browse tree improvements to potentiate usage.

As a minimum scope, “Collection drawer improvements” and “Collections quick access” will not be included. These need some maturation and are scheduled for a follow-up epic.

## Reference Documents

## Glossary

- DnD: Drag and Drop

# Functional Requirements

<https://app.prodpad.com/ideas/15/userstories>

## Non Functional Requirements/ Software Attributes

### Scalability

No remarks.

### Privacy

No remarks.

### Performance

No remarks.

# Features to Implement

## Requirements

*Drag and drop should be integrated with all listings (result layouts) that are provided by Nuxeo.*

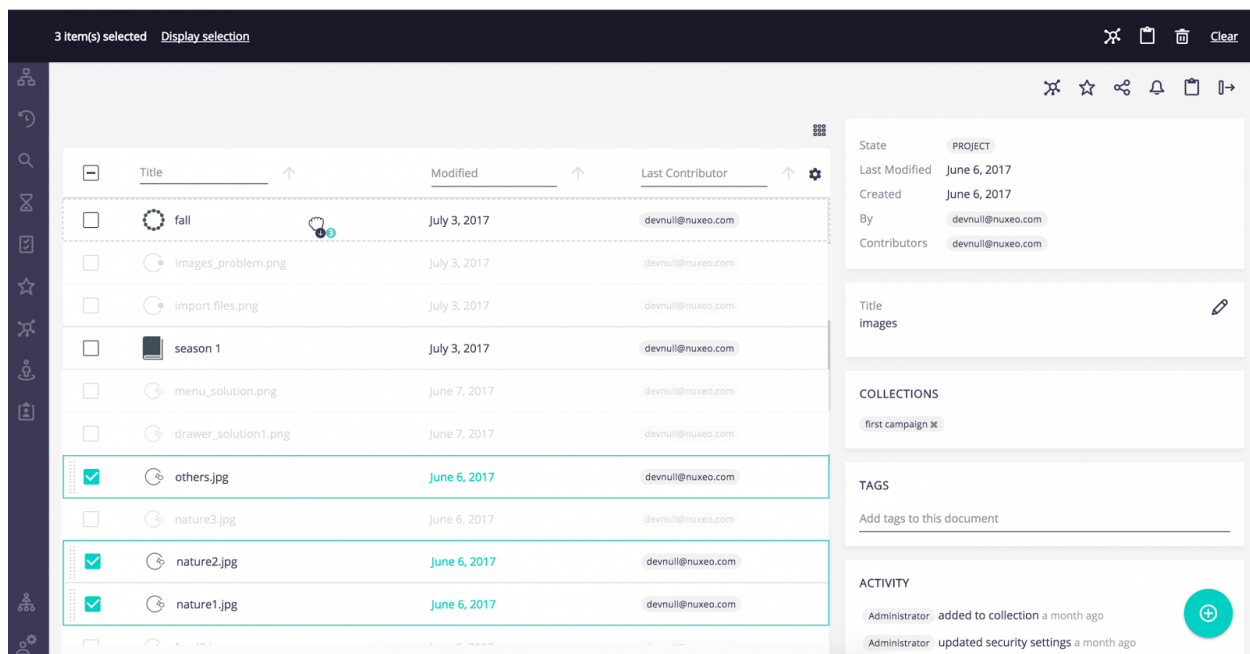
## Feature(s) description & Mockups

### Move to containers

On a listing, after selecting assets, if the list supports DnD, it should display an icon on the item's left to convey that information to the user.

If a user grabs one of the items, the icon changes to a grab with a badge to indicate that we are in DnD mode and to enumerate how many assets will be acted upon.

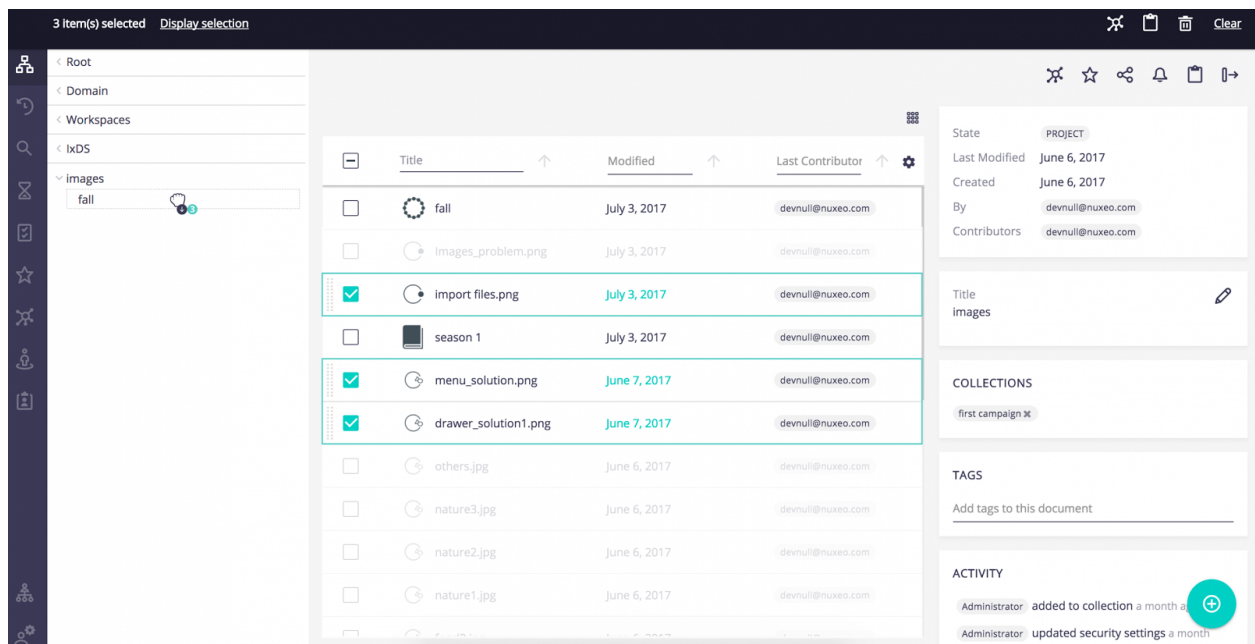
Hovering a container will change once again the icon with a down arrow, indicating that assets would be dropped there, moving them. The hovered container also highlights with a dashed border to make it clear on which asset we are acting on.



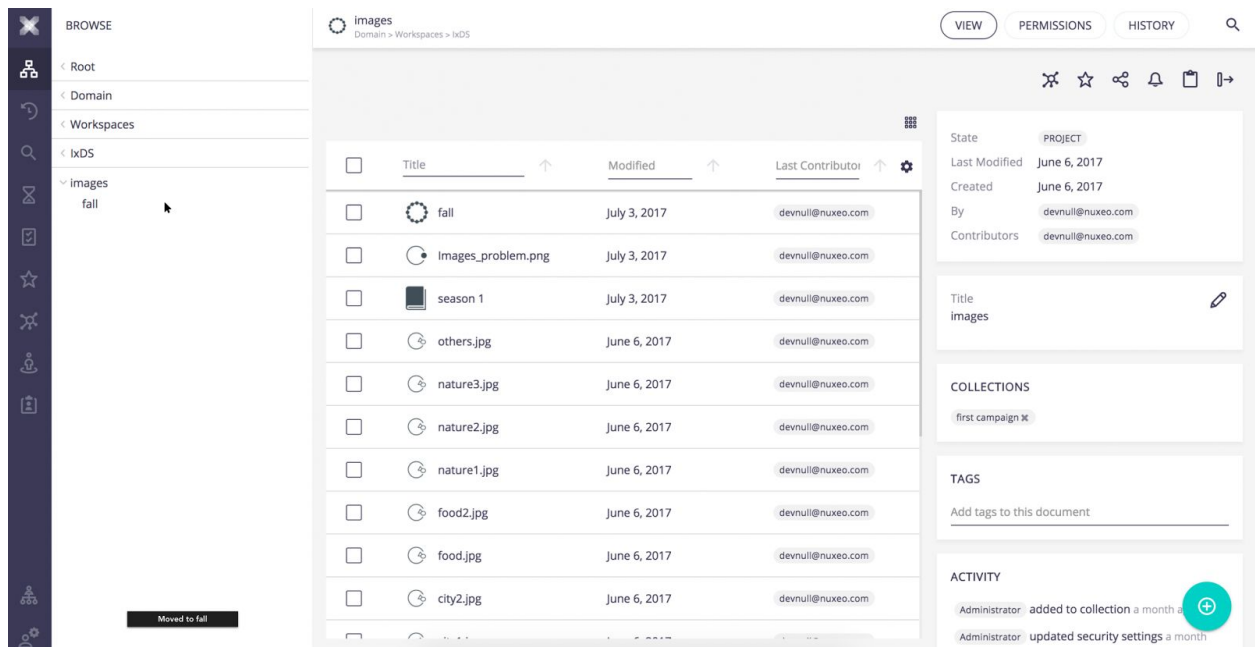
As not all relates just to the displayed listing, user should be free to move assets to other locations. The document tree and all browse drawers should allow for DnD.

By now, all Document Tree ancestors are not openable and navigate to the correspondent container on tap. This behavior should change and the user allowed to tap on the arrow only to open that node. This will allow the user to navigate to any part of his content without changing main content and, therefore, the selected items.

After grabbing assets, the Document Tree can be used as a drop area. Hovering a node on the tree will provide the same feedback as mentioned above. And dropping the assets will result in their correspondent relocation.

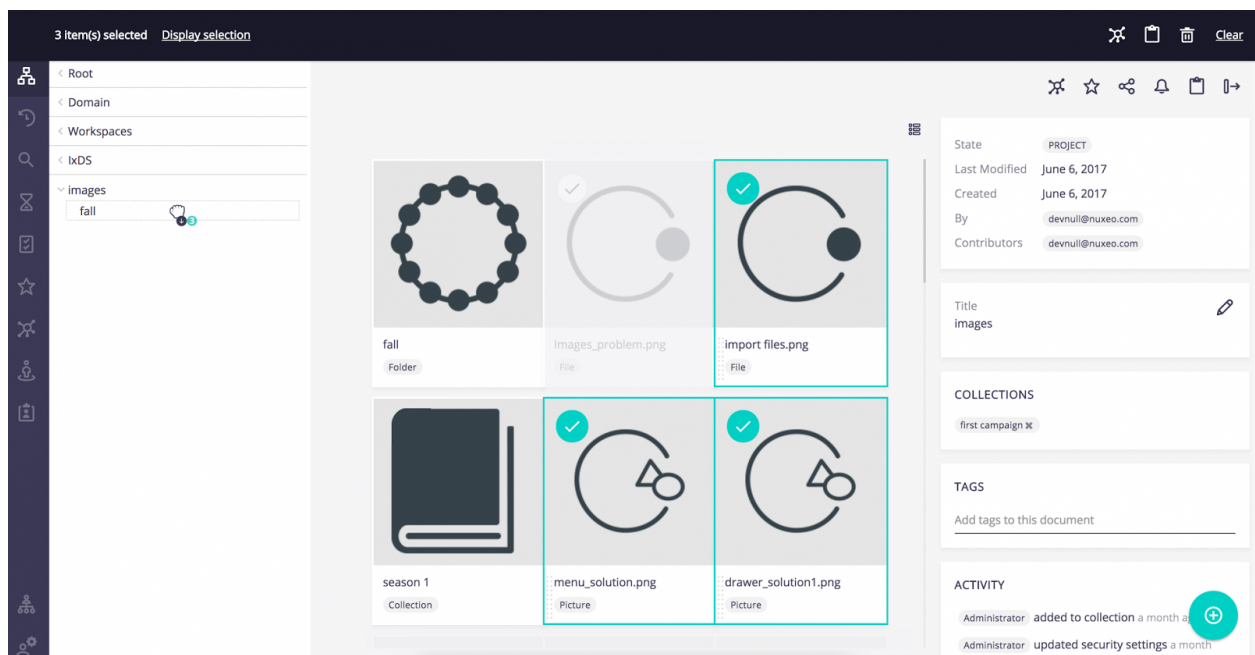


After DnD, if the operation is successful, a toast will be presented with a success message.



In case the operation fails, the toast design system should provide that information as it does in other cases.

Other listing types should have exactly the same behavior as shown below:



This feature alone should allow user to move assets to any place on the repository.

Note that a user can open any other browse drawer page (with different root) or personal workspace, in order to move assets there.

## Favorite queue

On special case it the Favorite drawer page. Content on its queue will be similar to any other listing in terms of dropping functionality. All container will be drop targets with same type of feedback.

This will allow users to use favorites as dropping flow when they need to move assets quickly to several destinations.

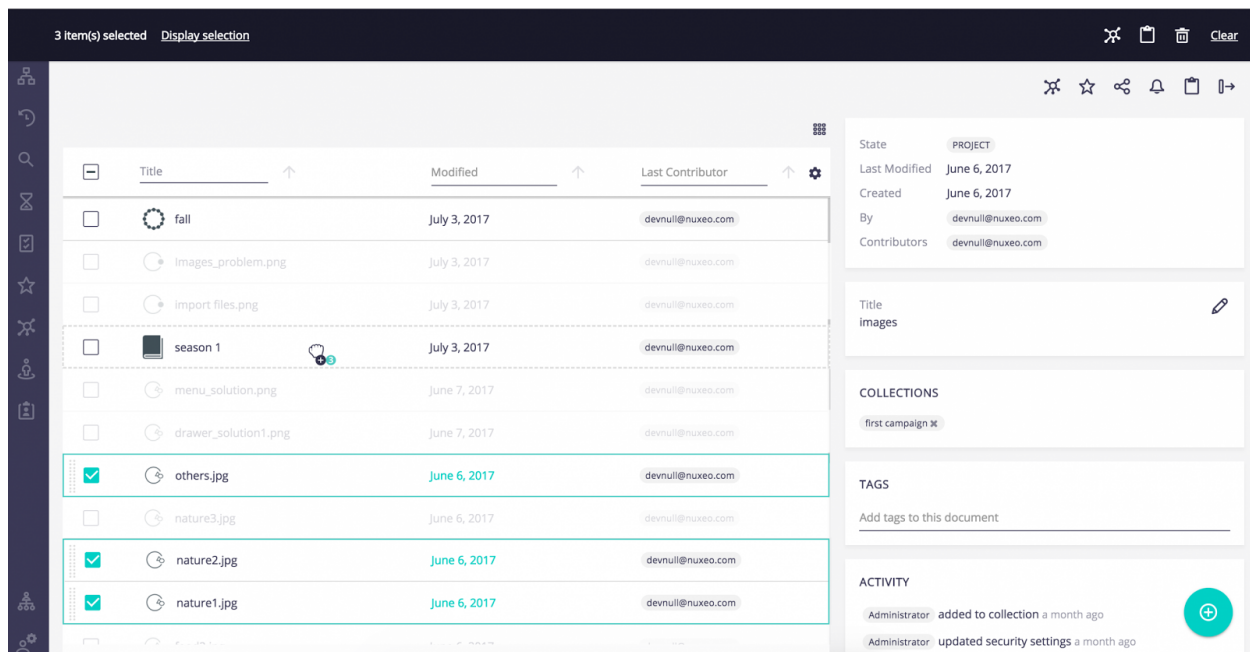
## Add to collection

Dragging and dropping to a collection is a different use case. On collection, there is the possibility to have an asset on different collections at the same time. They may be used to group assets in a way that is sharable and mutable.

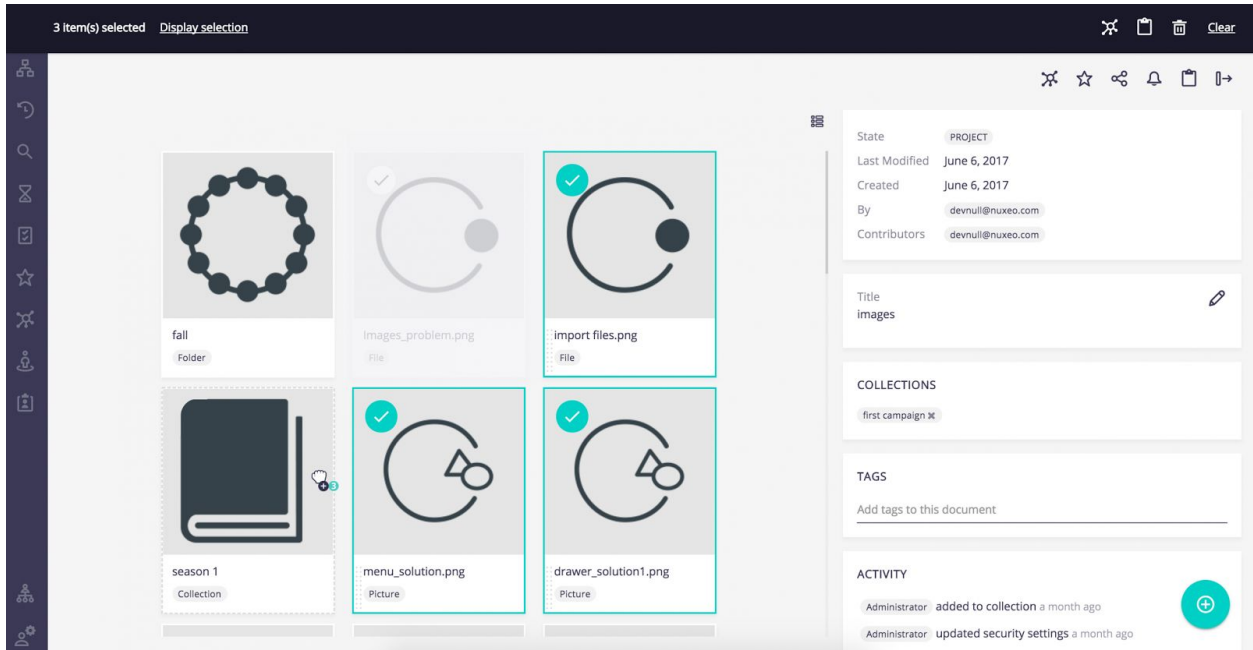
When a document is added to a collection it doesn't relocates its location on the repository.

As such the feedback need to differ to accommodate this difference.

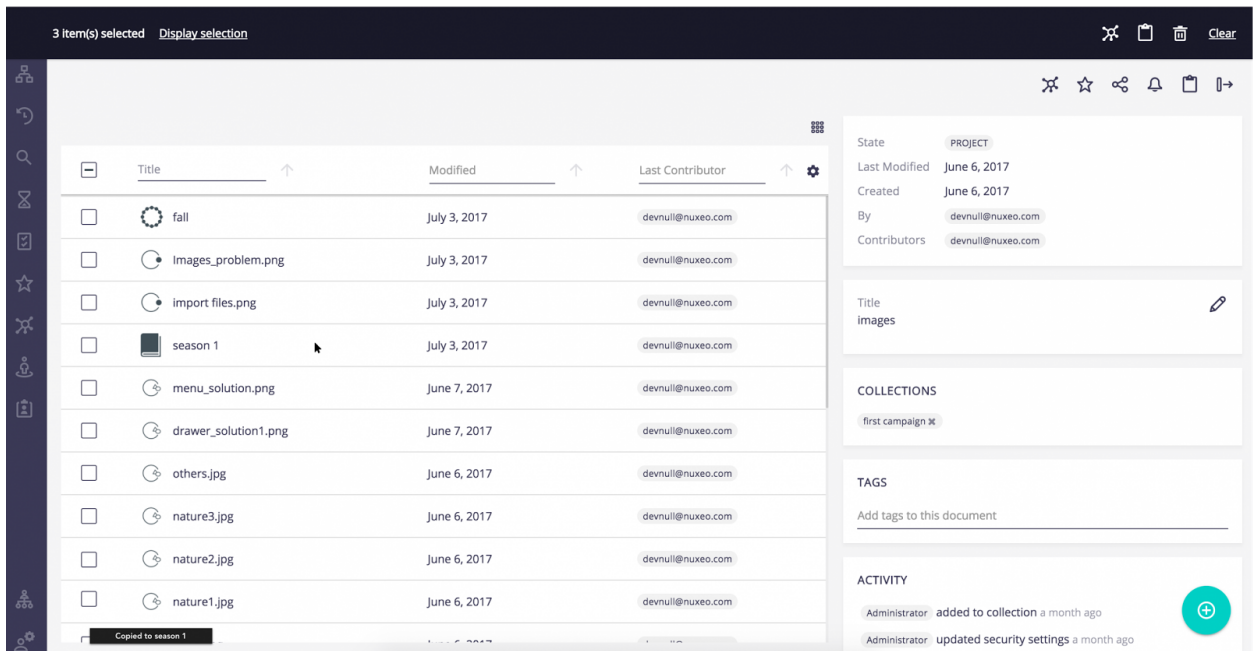
When hovering a collection, the down arrow is changed to a plus sign.



Like with folders, behavior should be similar with different result item types.



Success feedback should use the same system.



## Collection drawer improvements

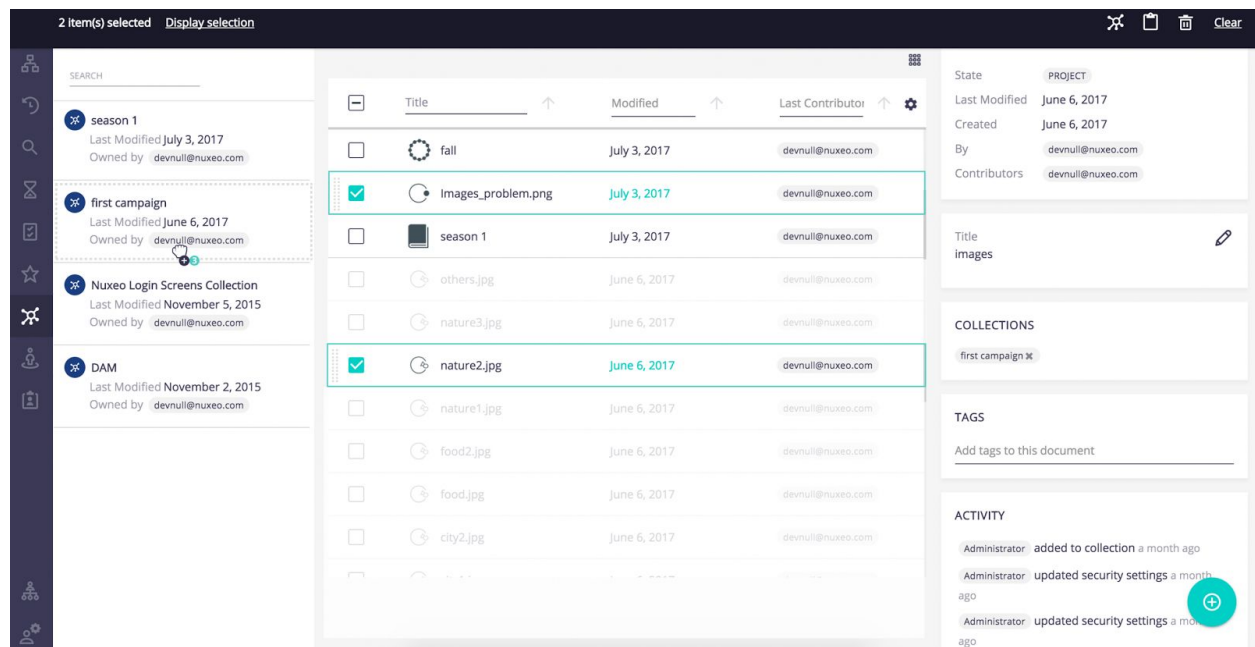
Collections are a really important use case and is common for an asset to be added more times to collections than moved from container to container. This happens because collections are

more volatile and mutable and, even sometimes, used on a time cycle (eg. to define pictures for weekly newsletter).

This makes finding and DnD assets to a collection a prime user story that needs its own improvements.

To accommodate the former user flow, the collection drawer page should have a text search on top that allow filtering to fast search a collection. This will enable the user to find and use the correct collection very efficiently.

The filter will be based on title and when some text is introduced, a create button will appear to allow the user to use the search as a collection name.



## Collections quick access

Furthermore, the usage of collections is linked to search. Searching for assets to group in a collection is a common use case and drag and dropping can improve this flow.

After a selection and when starting dragging, the top selection bar should increase height and display a set of collections where the user can drop to add.

[specify how this set is gathered]

## Generic Drag and Drop

- Listing (including used ones on pagination control topic), both result list, container content and queues, as well as Document Trees should allow user configure DnD capabilities when adding them to contributions or layouts.

All designs at <https://app.prodpad.com/ideas/15/mockups>.

## Limitations

No remarks.