



Nuxeo

Nuxeo

Asset layouts - minimum scope -
Specifications <review mode>

Last update:

Table of Content

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

Feature Overview

Goals

Enable asset layout configuration with a straightforward and simple process, allowing them to designer layouts they require.

Allow users to edit any type of assets without complexity and current layout restrictions.

Reduce complexity on the code side

Provide a full page layout configuration and not just layout areas configuration.

Problems to solve

Static layout areas : When creating new document type, the process of defining View, Metadata, Edit, Create and Import layouts can be time consuming and repetitive. Users are stuck with a default static asset page with predefined areas for configurable layouts.

One layout template doesn't always work : Different asset types have different ideal views and each asset type may have multiple useful views: a static page structure is not compatible with this.

No layout granulation : When configuring the complete document page layout, it can be messy to defined the metadata fields and the page structure at the same time.

Short element reuse and relation to metadata : Currently, there is no easy way to reuse created layouts or even layout elements. User as to create everything from scratch or use code editor to introduce project elements - which don't have any relation with schemas, facets or document types -. This issue makes it troublesome to produce groups of metadata layouts.

Edit form on a static, confined placement : Also, edit layout is, by now, very narrow in terms of space and flexibility. Users have a mandatory area where all edition is done. And there isn't an easy flow to create and (re)use groups of metadata to edit. Still in this issue, edit form is done inline, not in a zen mode, which add complexity and noise to user experience.

Assets has same layout to every user profile : Asset pages and their layouts are equal to all users, user groups, in all contexts. The only possible difference is if the user has read permissions or not. Nevertheless, depending on the user role, the optimized asset page should be different. Even, the edit pages (like Create, Import, Edit) should be different depending on the function of the user.

Hard to create alternative views : Finally, some asset types need to have more than one view page. This can be defined today with document pages, but that process is targeted more at transverse pages that make sense in the scope of all or some asset types. This configuration type, also doesn't take advantage of all the layout configuration features.

Audience

- *target audience*

Scope

This is a minimum scope that does not target all identified problems. Instead, it proposes a short timeframe solution that already solve some critical issues going a direction compatible with the broader scope.

Issues it proposes to solve:

Edit form on a static, confined placement

One layout template doesn't always work

Short element reuse and relation to metadata

~~Static layout areas~~

~~No layout granulation~~

~~Assets has same layout to every user profile~~

~~Hard to create alternative views~~

Reference Documents

Glossary

- Term A:
- Term B:
- Term C:

Functional Requirements

User Stories:

Non Functional Requirements/ Software Attributes

Scalability

Privacy

Performance

N/A

Features to Implement

Requirements

N/A

Feature(s) description and Mockups

On this scope, features will be restricted to layout change and edit mode improvement.

These improvements and changes are part of a bigger scope that will mean structural changes both to web UI layout engine and Nuxeo Designer asset configuration tools.

In order to have a minimum scope that: made sense on a broader sense and that could with some extension test this broader scope; we defined these features as an implementation of what would be possible with the larger view.

To this effect, features presented here have the global asset topic as a starting point and use systems and mechanics from that work.

We had created several template systems, where the user could add and remove metadata groups. These groups could be visible just to some users. Therefore, all template systems needs to act in an elegant fashion independently on the number of metadata groups and need to scale to allow any undetermined number of this groups.

These template systems should be present in future in Nuxeo Designer so when configuring a new document type, the user has some interesting starting points.

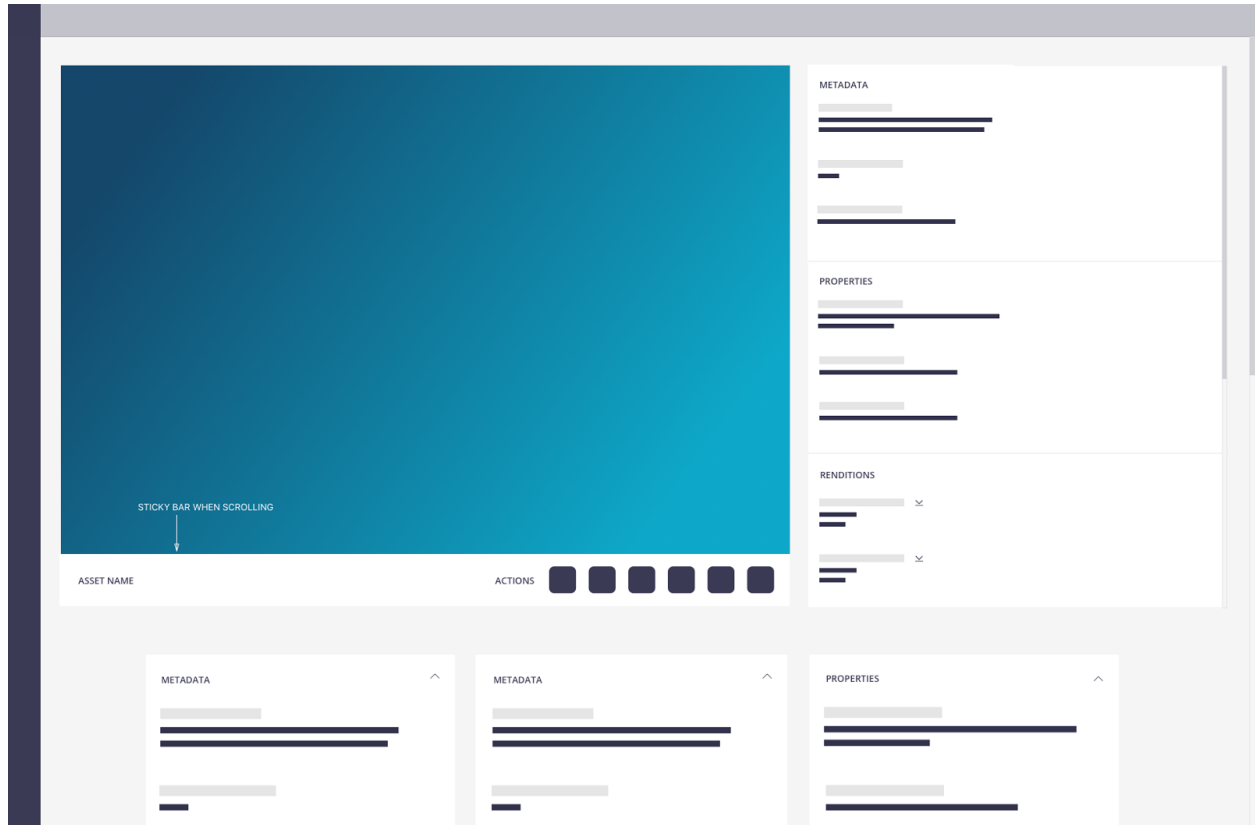
As a minimum scope, this will not intent to define these template systems nor any means to configure them, but instead, to use these templates to redefine Web UI default view page layout to: test this new approach and to allow the metadata layout to better scale, while improving UX/UI of asset's view page.

Previewable Layout Template System

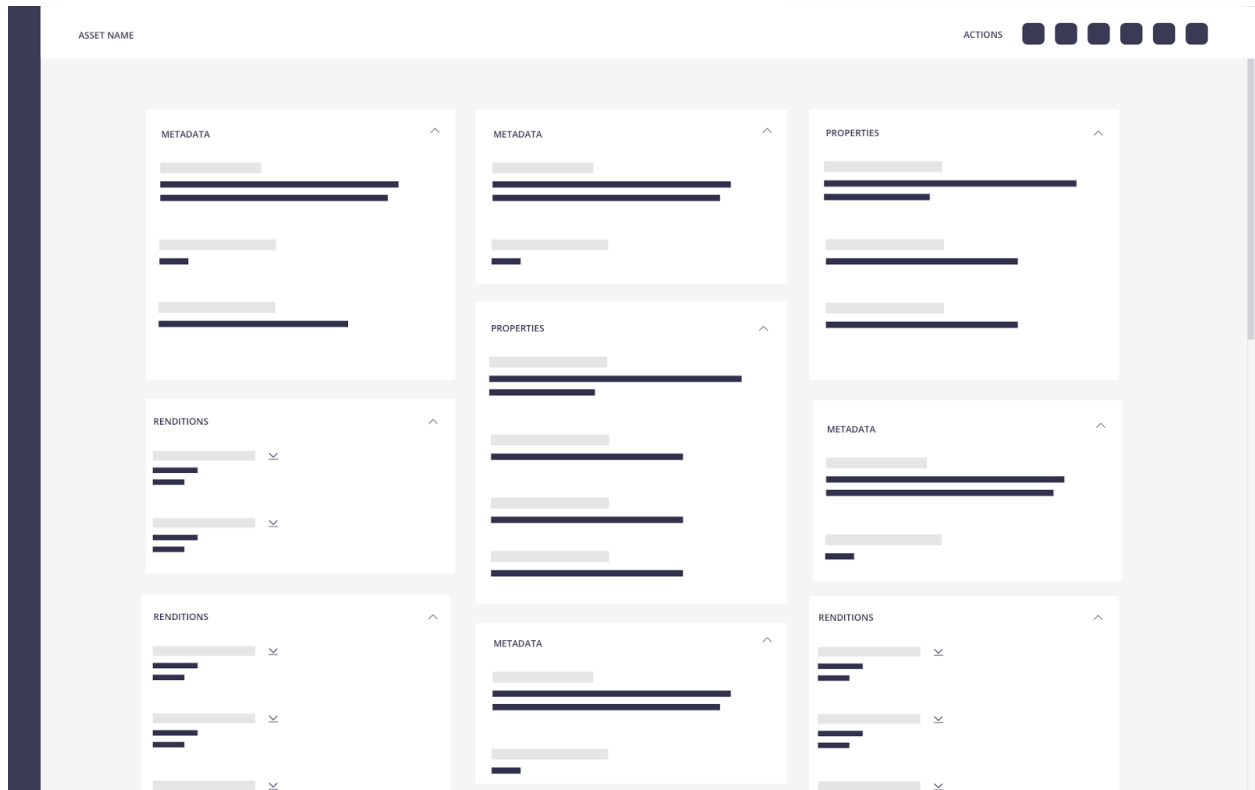
The first used layout system tries to target assets that have a main previewer content (e.g. pdf, image, graph, 3D model).

This layout system is composed of 3 main areas:

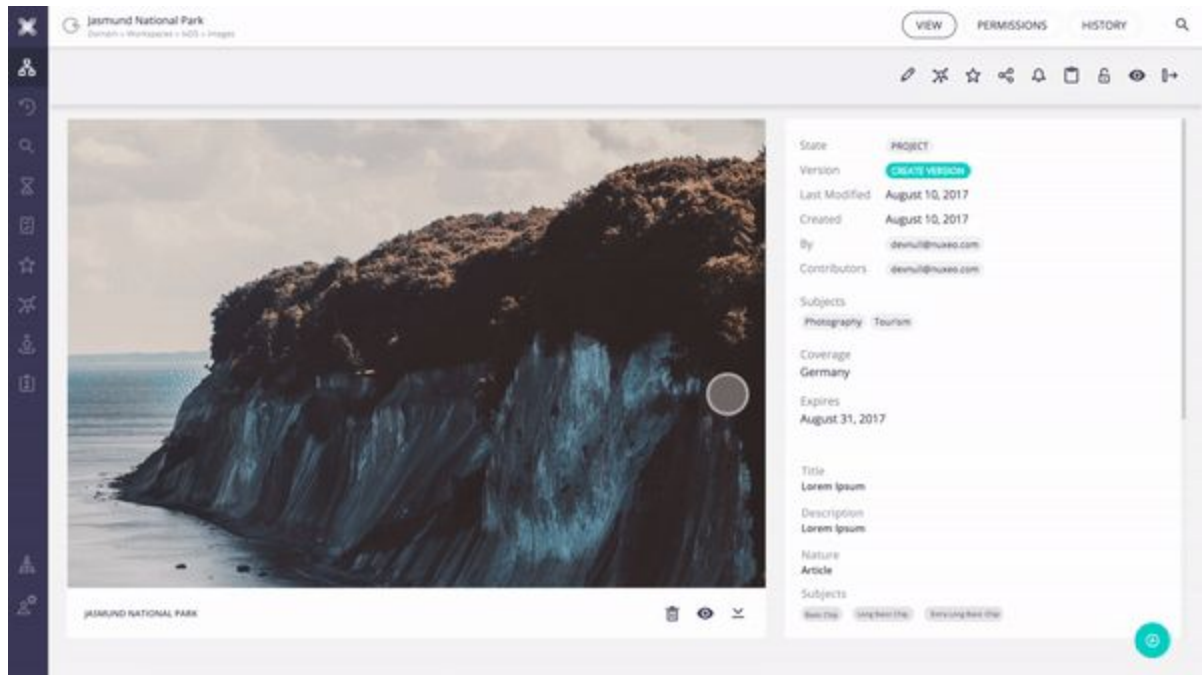
- **Preview main area** : (top left) Static area that can only have one group. This area takes almost the whole page height. Leaving just enough space so that the user can scroll easily - in our current case, the view layout -.
- **Main metadata** : (top right) Static area that can take any number of metadata groups. This area take the same height as the preview area. When its content overflow a scroll bar is presented. This will enable all the preview related metadata to be presented side-by-side with the preview.



- **Complementary info** : (bottom) This area uses a masonry style arrangement to allow reconfiguration and adaptation change size and changing number of groups presented. This area will continue indefinitely as the user scrolls down on the page.



When the user scrolls down, the asset action bar will stick on top. If the user continues to scroll it will disappear, but appear immediately as the user scrolls back slightly up wherever the position on the page. Below an example of this behaviour :



This template system will be the bases for File, Note, Picture, Audio document types.
Here is the high fidelity mockups and description for File.

File

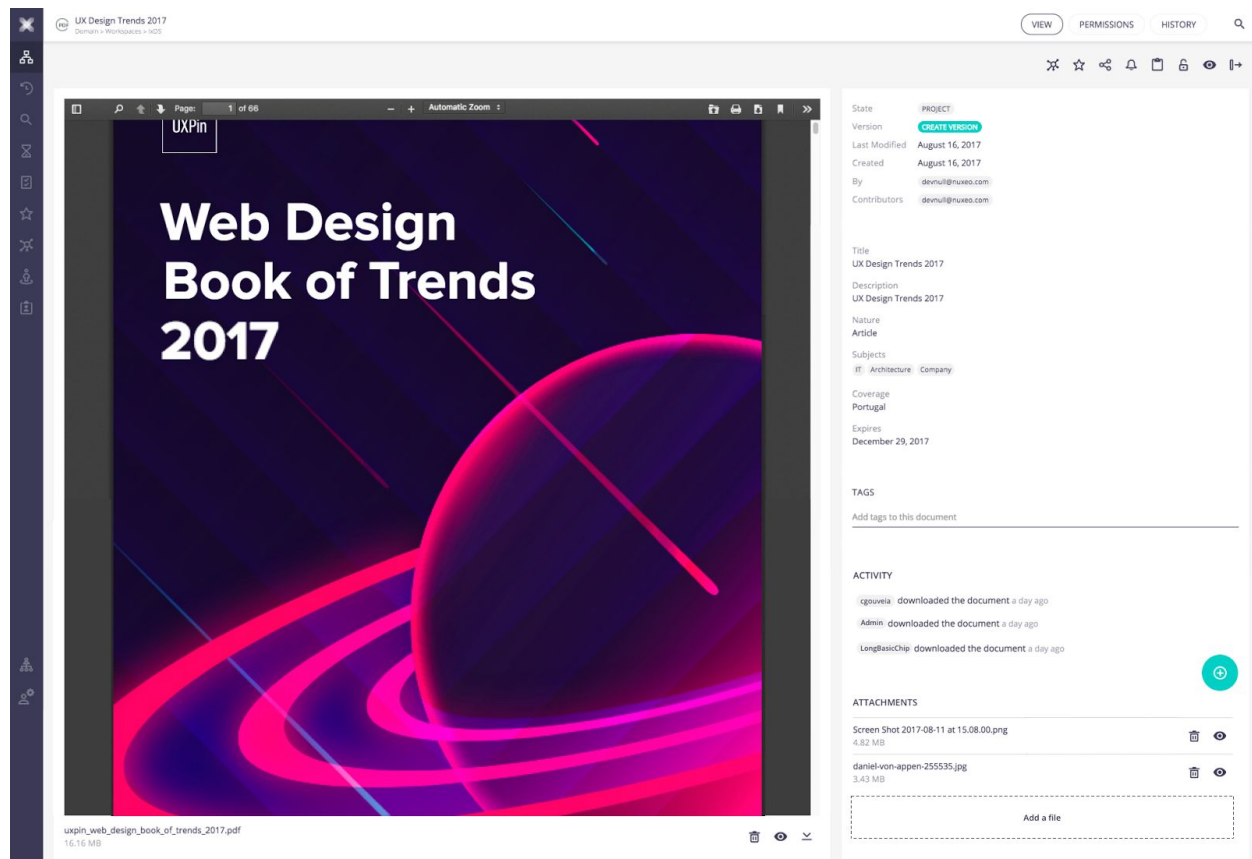
For the File document type, the file content is the main information to preview.

Preview main area will be filled with the current main blob preview, title and actions that is shown already in web ui, basically, the view area without the attachments.

All other metadata groups will all be added to main metadata area. These include :

- Main details (author, dates, etc).
- Metadata layout (configured by user)
- Tags
- Activity
- Attachments

For this scope, it wasn't added any other metadata groups to the original layout, so there will be no elements on Complementary info area.



On small screens, this layout will collapse in just one column.

Picture

On Picture document type, the same template system is used.

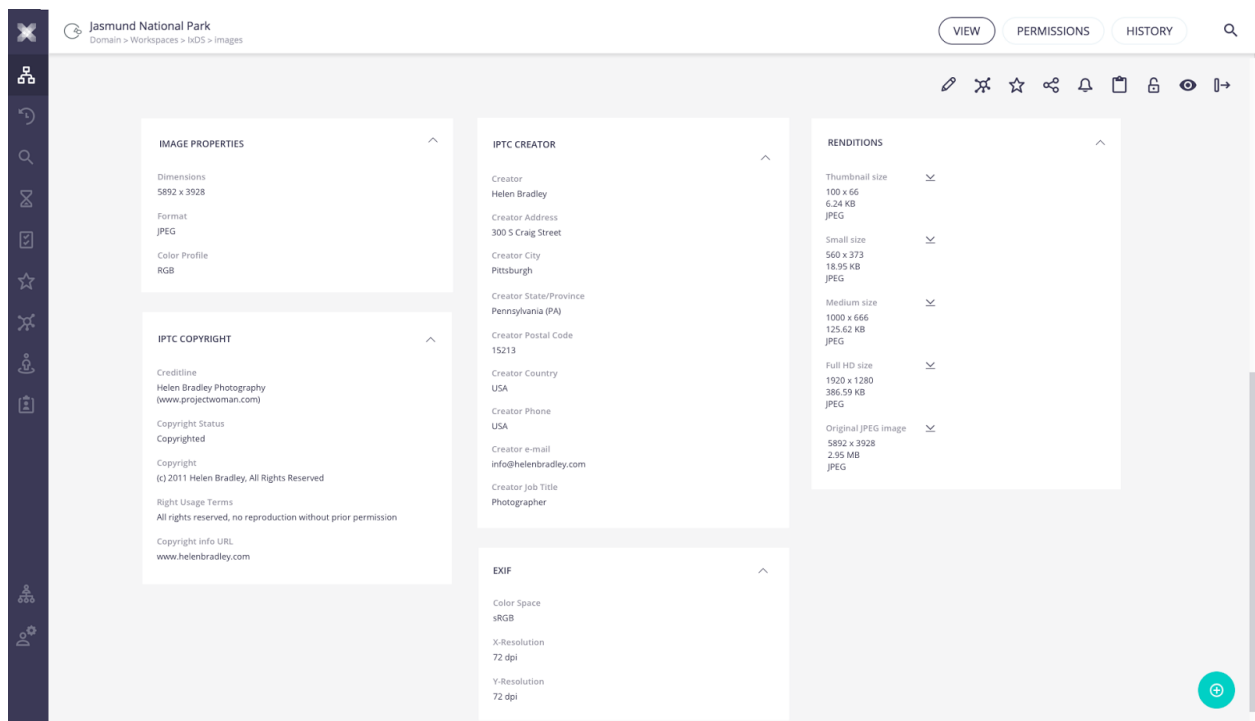
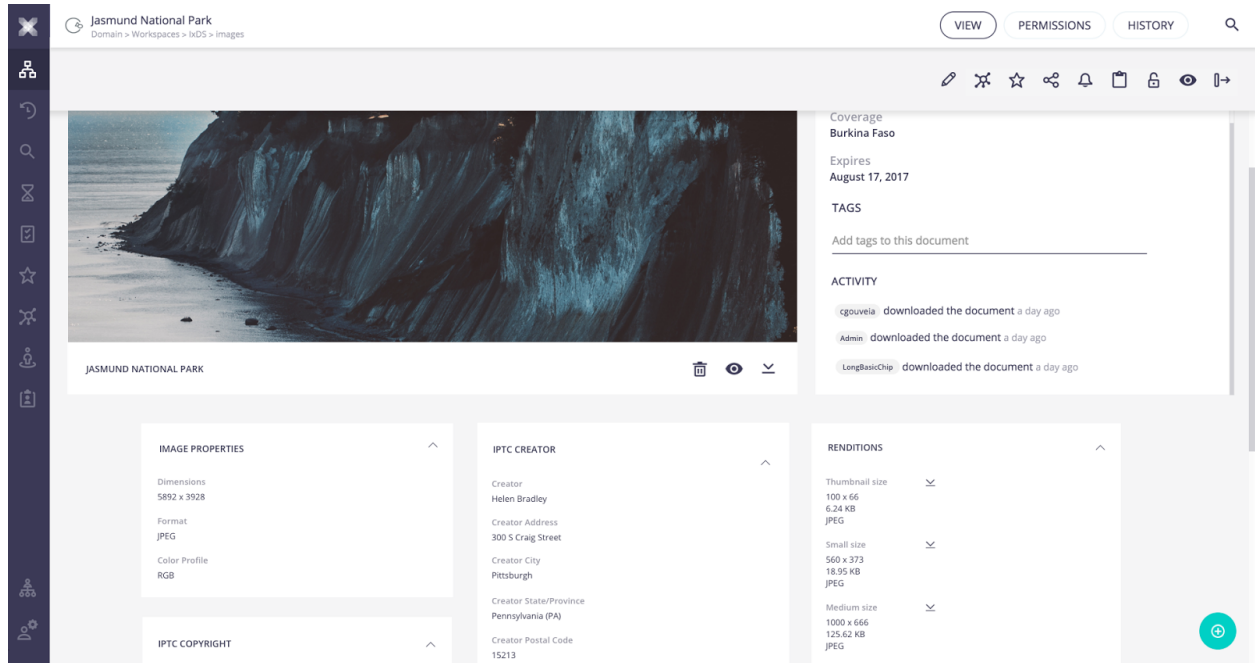
On the main preview area will display the image. The main metadata area will present the same info as file, but without attachments - as they don't exist here -.

A big difference is the extensive usage of the complementary info area. This will be used to display all complementary information on the picture, like exif, iptc, renditions or image properties.



Subjects

Thumbnail size	▼
100 x 66	
6.24 KB	
JPEG	
Small size	▼
560 x 373	
18.95 KB	
JPEG	
Medium size	▼
1000 x 666	
125.62 KB	
JPEG	
Full HD size	▼
1920 x 1280	
386.59 KB	
JPEG	
Original JPEG image	▼
5892 x 3928	
2.95 MB	
JPEG	



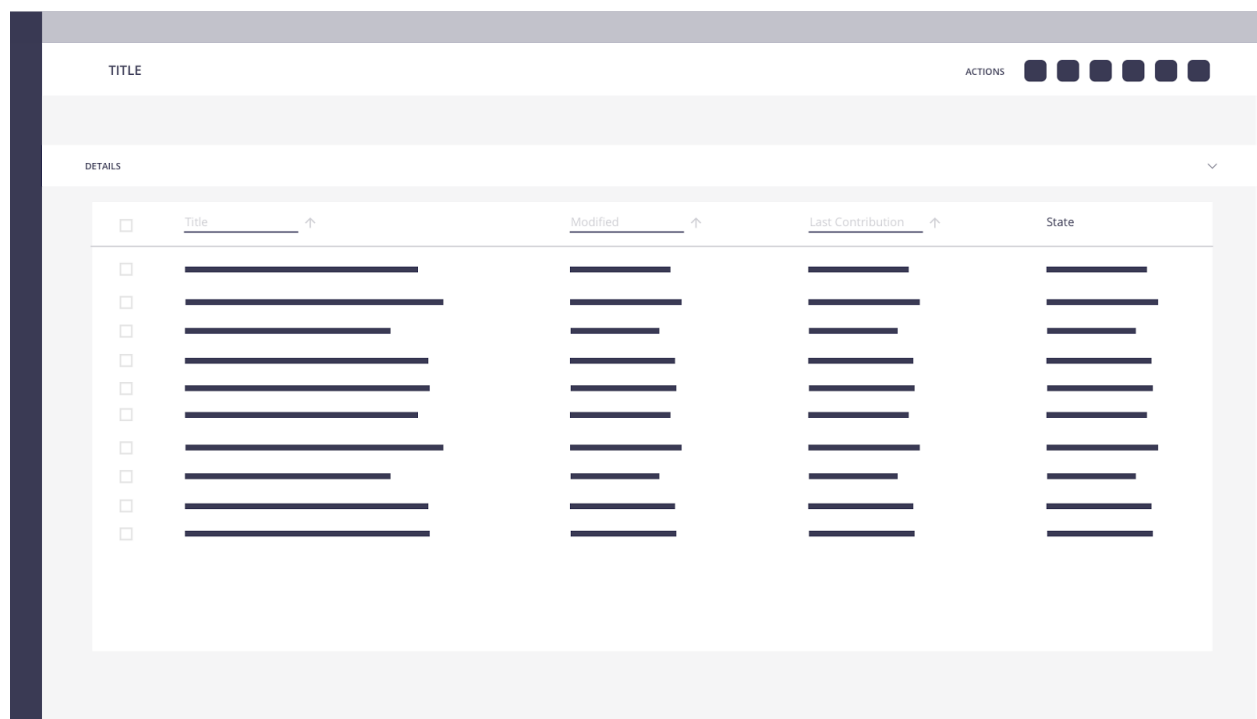
Above there are several high fidelity mockups that present behaviour on different scroll situations.

Container Layout Template System

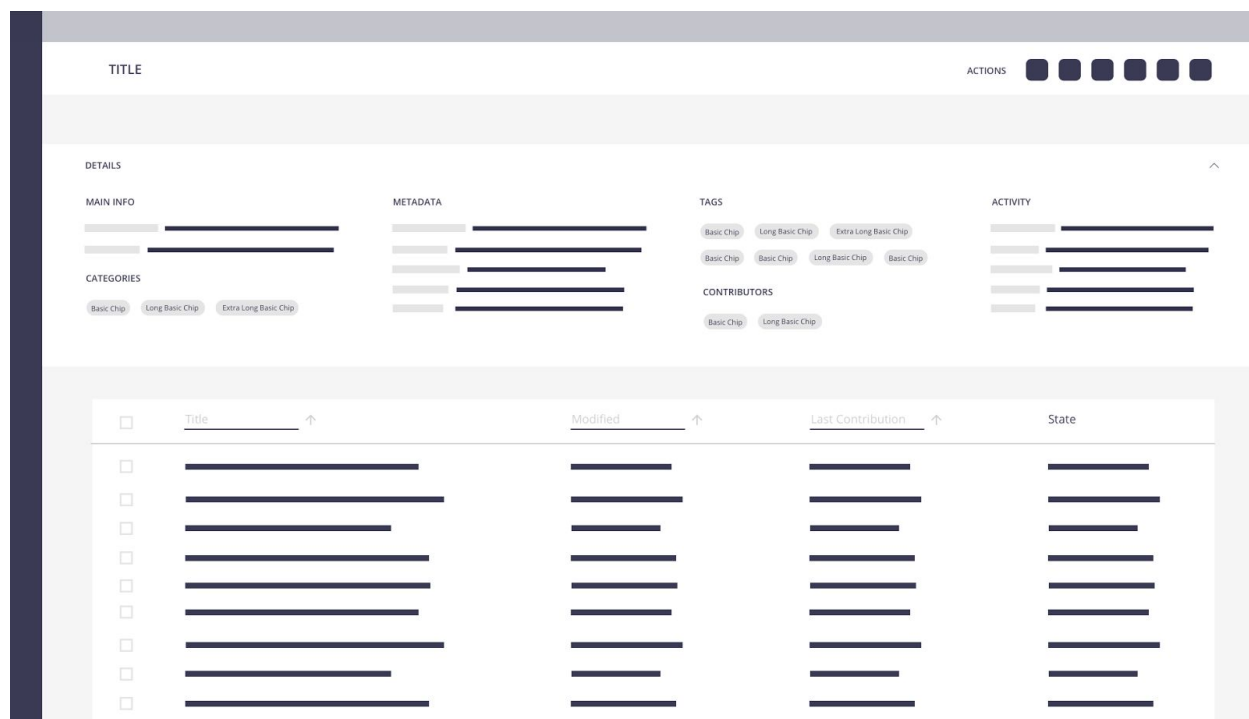
For document types that focus on containing other assets, like folderish or collections, another template system was design to better take advantage of this use case.

This template will have two main areas:

- **Detail metadata** : this area will be collapse by default and will accommodate all metadata groups.
- **container** : the container area will be a static area the will serve to allocate the contained assets. It is full with to allow several columns in table mode and uses most of the screen to present a big number of assets - in our current case, the view layout -.

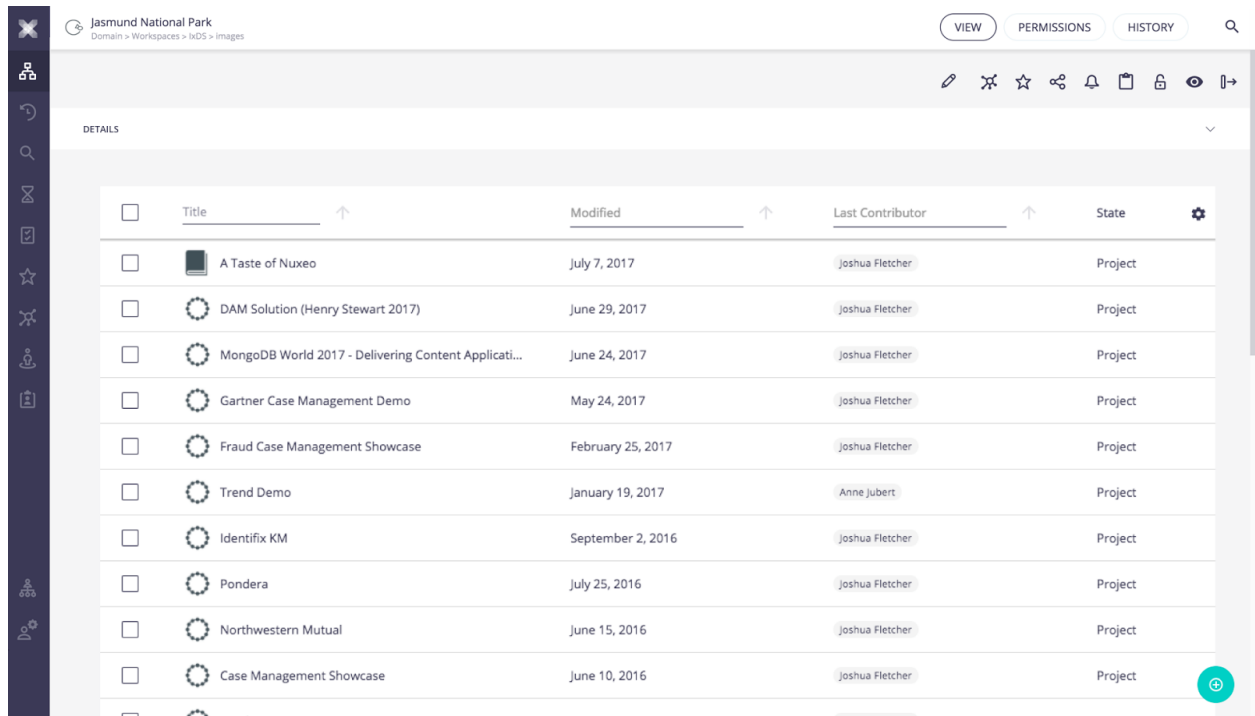


Above is presented the template in its initial form and below, when user expands the detail area.

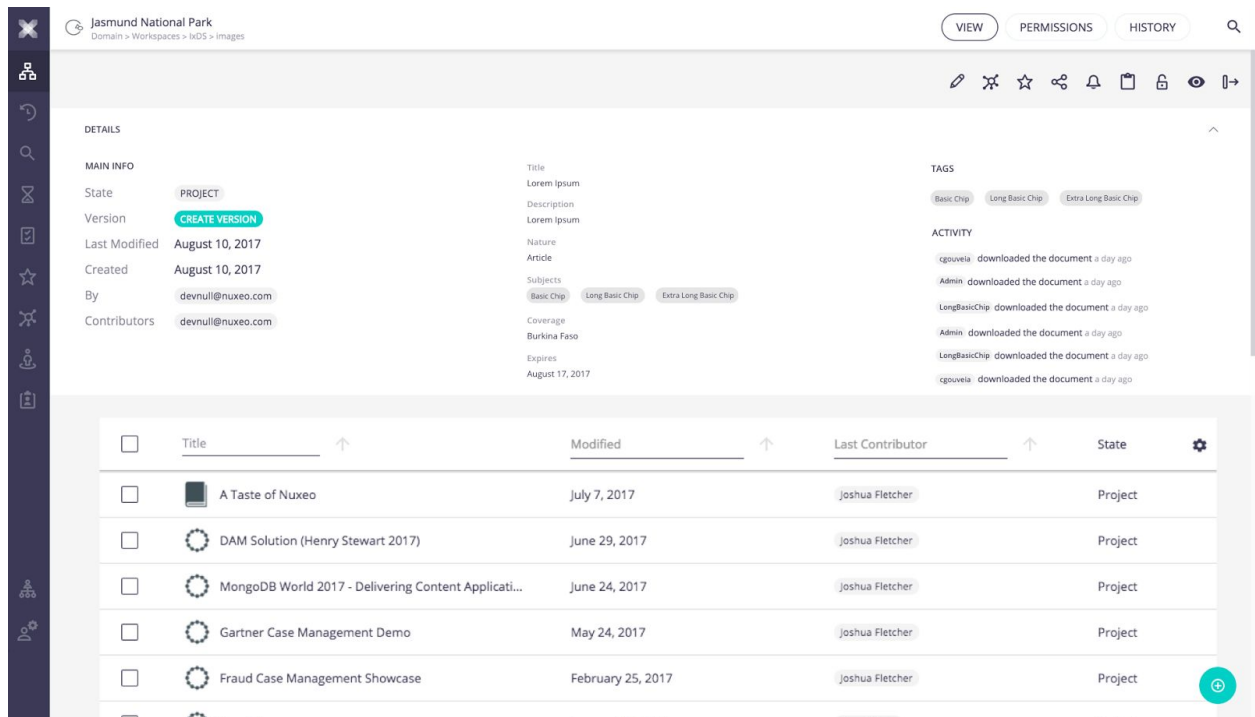


Folder

As an example, it is presented the high fidelity mockups for Folder document type. Notice there has been some improvement since the original layout template. One of the changes was to display only 3 columns on details to better work with smaller screens and with the open drawer.



Above, it is noticeable that contained assets really take most of the page real estate. If the user needs to investigate any metadata details on the container asset, it is possible to expand the details area. It works with a masonry system that enable any size of metadata groups.



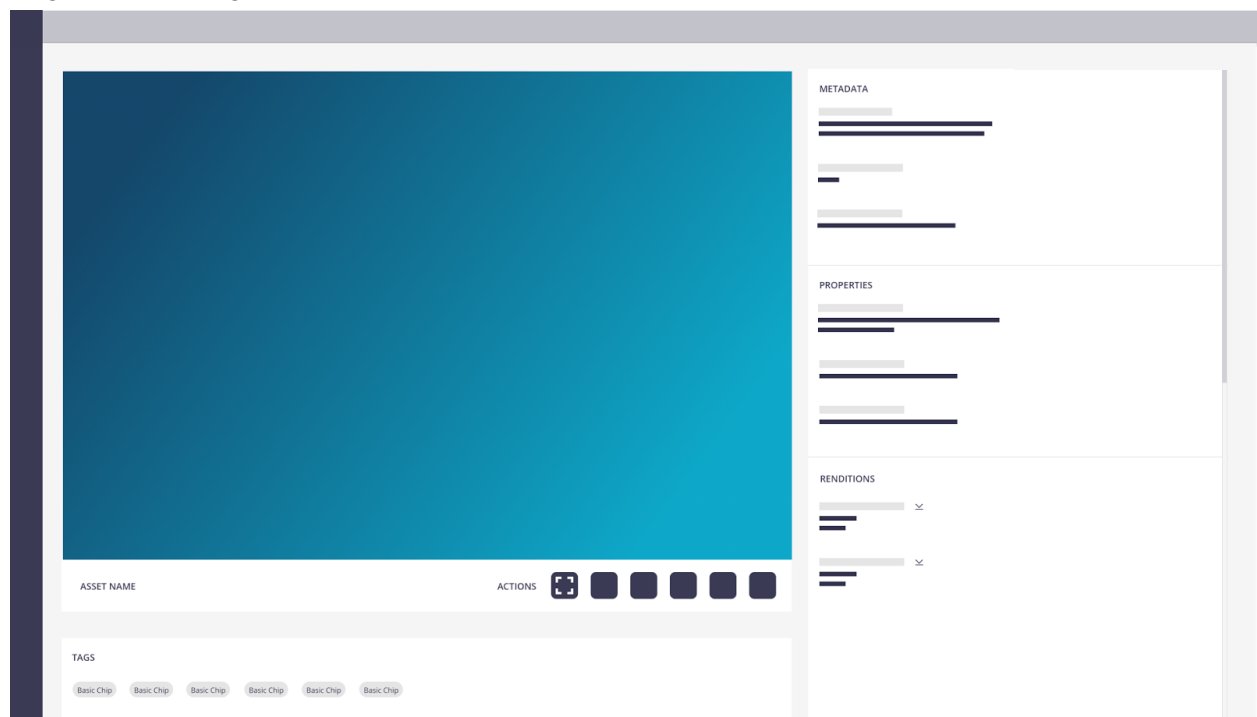
On Folder and Collections it will have the following groups:

- Main info
- Metadata layout
- Tags
- Activity

Asset actions will be displayed on top, will disappear on scroll down and immediately appear on scroll up.

Wide Preview Layout Template System

Pictures and file preview makes sense occupying a big part of the screen and it's full height (eg. pdf with full pages). But other content type don't require this and a smaller area is enough to display the preview properly to the user. So this template system set a static preview area height. Depending on the screen size, this allow users to be presented with more metadata.



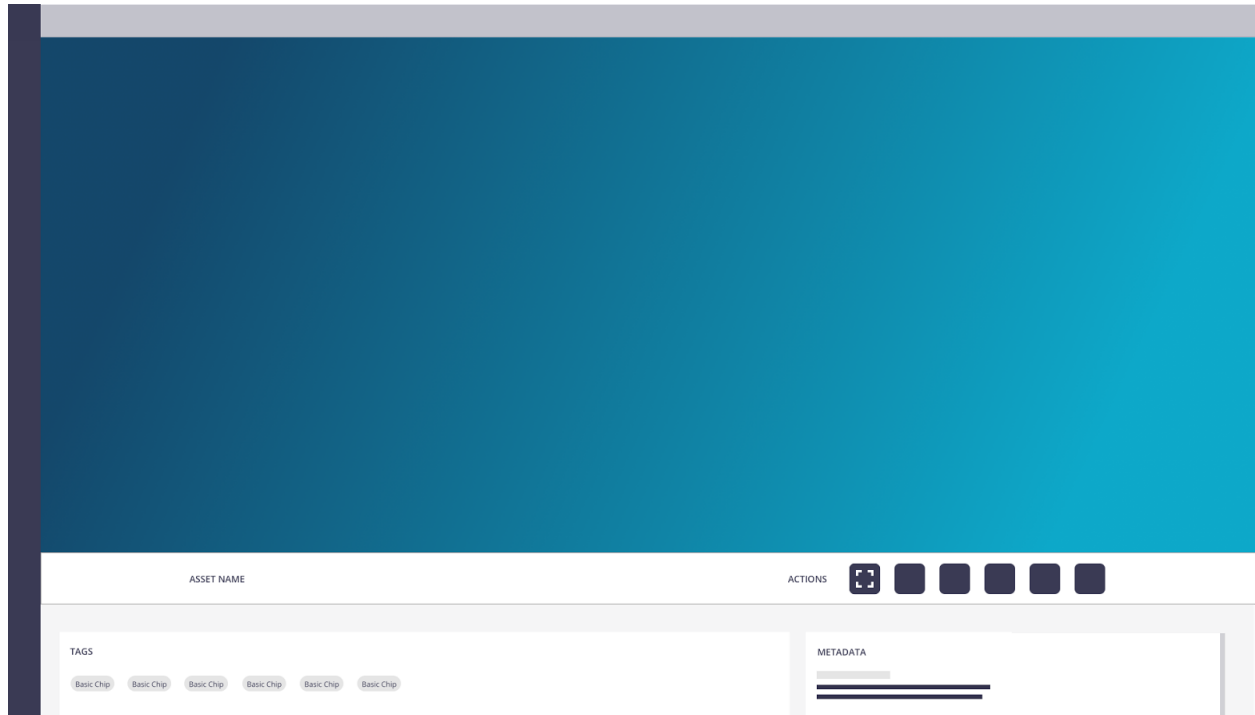
Like Previewable Layout this template also have the same 3 areas, but with slightly distinct behaviour. Main metadata area on the right side goes all the way down until there isn't any other content on its left. Nevertheless, this area should have a minimum height set.

The preview area is also static and takes only one metadata group (eg. preview and blob actions) but the difference is that it has static size in not dependent of screen size.

The complementary information area is present only in the left side and has only two columns. It is located below the preview area.

As the preview is restricted to a smaller area, we can use the full width of the screen for a bigger preview. It is especially fitted to videos, but can work with other content that width preview makes sense.

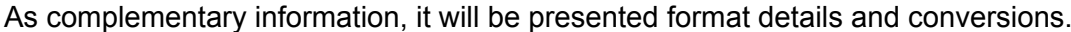
To this effect, the layout has the wide mode that can be dispatched with a blob action.



When on wide mode, main metadata areas will be place itself beneath the preview area. On scroll, asset actions have the same behaviour as in previous template system.

Video

This template system is particular suited for videos as they tend to be wide. Just as in file, the main metadata are has all the same groups.





Edit action

<todo>

- Asset action
- Zen mode

The screenshot shows a web application interface for editing metadata. The top navigation bar includes the site name 'Jasmund National Park', a breadcrumb trail 'Domain > Workspaces > XDS > Images', and buttons for 'VIEW', 'PERMISSIONS', and 'HISTORY'. A left sidebar contains various icons for navigation and actions. The main content area is an 'EDIT' modal window with a close button in the top right corner. The modal contains several sections of metadata fields:

- Title:** A text field containing 'Lorem Ipsum' with a red asterisk indicating a required field.
- Description:** A text field containing 'Lorem Ipsum'.
- Nature:** A dropdown menu currently set to 'Article'.
- Subjects:** A section with three chip-based tags: 'Basic Chip', 'Long Basic Chip', and 'Extra Long Basic Chip'.
- Coverage:** A section with a dropdown menu set to 'Burkina Faso' and an 'Expires' date field set to 'August 17, 2017'.
- TAGS:** A section with the label 'Add tags to this document' and an empty input field.
- IPTC COPYRIGHT:** A section containing fields for 'Creditline' (Helen Bradley Photography), 'Copyright Status' (Copyrighted), 'Copyright' ((c) 2011 Helen Bradley, All Rights Reserved), 'Right Usage Terms' (All rights reserved, no reproduction without prior permission), and 'Copyright info URL' (www.helenbradley.com).
- IPTC CREATOR:** A section containing fields for 'Creator' (Helen Bradley), 'Creator Address' (300 S Craig Street), 'Creator City' (Pittsburgh), 'Creator State/Province' (Pennsylvania (PA)), 'Creator Postal Code' (15213), 'Creator Country' (USA), 'Creator Phone' (USA), 'Creator e-mail' (info@helenbradley.com), and 'Creator Job Title' (Photographer).

At the bottom of the modal are two buttons: 'CANCEL' and 'SAVE'.

Limitations

This is a minimum scope approach that does not solve all problems.

