



Nuxeo

Web UI

Pagination Control Specifications

Last update: 12/07/2017

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

Currently, document list enable an infinite scroll mechanism that allows users to view documents in linear order without having to select a page.

This approach has several problems: it is not possible to share a specific page and getting a document takes linear time regarding its position.

Other solutions like pagination links may be troublesome as it make navigation in pages slow and it is never clear to the user the importance of each page or what it might find in them.

Common problems that exist with pagination in listings are:

- No quickly access list positions
- No information on listing content segmentation

Users should be able to navigate on the total set of a list. Be able to share a direct link to a part of that set and see aggregations of the sorted field on the side of the scroll bar.

On the other hand, current result list do not enable bulk selection with shift nor sort ordering. This issues are also detailed by this topic.

Audience

Product, engineer and QA teams

Scope

Total list space navigation, selection and sorting specs as well as description on how new list items can be contributed.

Reference Documents

<https://photos.google.com/>

Part of the inspiration for this topic

<https://doc.nuxeo.com/nxdoc/page-providers/>

Nuxeo page providers documentation

<https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-from-size.html>

Elasticsearch limitation on total documents that can be retrieval

Glossary

- Aggregation: Elasticsearch function that aggregates results by a field over a determined method
- Bucket: A singular item of aggregation with the same value on the field aggregation

Functional Requirements

User Stories and Acceptance Criteria

<https://app.prodpad.com/ideas/7/userstories>

Non Functional Requirements/ Software Attributes

Scalability

Scalability may be a risk in this topic and a prototype should provide details how to workaround those.

Privacy

No remarks.

Performance

Performance is a risk in this topic and a prototype should be developed to assert it.

Features to Implement

Requirements

Aggregation segmentation requires Elasticsearch to be enabled on the page provider.

Feature(s) description

One very important aspect of this topic is the intention of results being reusable in Nuxeo UI Elements so project other than Web UI can take advantage of its features. Also, it should be flexible to allow several types of custom listings with custom items.

Full result scrolling

In order to allow for full result scrolling, all items should be available as placeholders and be load as soon as the are needed by the user. Specifics about how timing on loading new pages should be decided over the prototype.

To this effect, data for total result number should be generated with first pages already loaded. When a user scrolls, placeholder are shown for unloaded page items.

As soon as scroll stops, the on screen pages are loaded and presented.

Several thousands of items can be in a list, so it is relevant to use something very efficient like iron-list element.

This will result in initial items being correctly displayed, when user navigates, items will appear with information, and if he afterwards navigates to an already loaded scroll position, it will appear with data, with no need for loading.

The list element should allow for any type of layout. Several items per row or a row per item.

Today, there is a limit on `index.max_result_window` that is 10.000 by default and set the top limit for results on 'from + size'. In practice, this means that in a list with more than 10.000 (or the set constant) we are only able to show those 10.000 items.

There should be two approaches to this issue:

Review this variable and try to set the bigger possible number compared with negative impact.

When a list has more items that this variable, restrict list to this number and give feedback on this to the user (see mockups).

Aggregation segmentation

When navigation through the scroll the user only information is the total size of the results. He has to stop scroll to understand the current position context.

Nevertheless, Elasticsearch provides us with aggregations that in some restricted cases allow us to have enough information to segment total result.

This segmentation should be presented on the side of the vertical scroll bar. Each bucket has a horizontal line in the place it starts with the key text. This information should only appear when the user scrolls. On the scroll picker, it should be displayed by its side the current bucket.

If there are several aggregators that provide full result segmentation for the current sort, the side information should only display text on the aggregation with less buckets. The scroll picker should display text of the aggregation with more buckets (this should also be displayed as horizontal lines on the scroll).

Qualified aggregations

In order for an aggregation to qualify to segment the result space it should:

- Be ordered by the same field that the first list order
- Be order by key and not count
- 'Minimum doc count' ≤ 1
- Be sequential (eg. ranges are set by user and can have overlap or blank spots)
- Data Histogram, Histogram, Terms
- Have the complete result total in total

In terms of UI, the segmentation widget will appear when:

- Scroll is being used
- User pointer over right side of list

Segmentation will be hidden when scroll is over and user pointer is not over it.

Selection

Currently, selection is based on document ID's. Selection will only be available for loaded items. There are some cases of bulk selections that will have the following behaviour:

Select All - This feature, now works with the loaded pages only. As there is no way to present the user with an understandable way what pages are loaded, this functionality will be removed.

Shift Select - Shift select allow for bulk select from the last select item to the selected item. It should work when all the selected items are loaded. When the user presses shift and hover an item, it will highlight (the selection ticker) all the items that are going to be selected. In case, any

of those items are not loaded, this highlight wont happen and the shift select will result in common selection. (see mockups)

Bulk selection could be improved, but are out of this topic's scope. It would need some type of meta-selection not working directly with IDs.

Configuration in WebUI

As default, this new type of result listing should be present in search results, folderish layouts and drawer queues.

For each that will be default configuration. Queue will be limited to scrolling with no aggregation, selection nor sorting.

Search and folderish listing will fully enable all features. Nevertheless, specific configuration on new aggregations and sorting will be defined on the prototype stage.

Mockups

<https://app.prodpad.com/ideas/7/mockups>

Limitations

In terms of limitations:

- Elasticsearch does provide a limit on retrieval documents on a list. Nevertheless, user should receive feedback on it.
- Scroll aggregation info can only be presented with specific aggregation on the sorted field.

Features describe in this topic may not be feasible in non-regular items like justified grid and tables.