

# Generate preview on JCR-nodes that have changes

## I. Intro

---

Generate preview on JCR-nodes that have changes:

- POC requested  **MGNLUJ-5863** - POC: Generate preview on JCR-nodes that have changes **CLOSED**
- Mockup <https://zpl.io/VqJdXjm>

## II. What we are having

We have CMS filter chain which does the page rendering [Filters-CMSfilterchain](#). RepositoryMappingFilter, AggregatorFilter are mainly affected to the renderer process which is raised in RenderingFilter.

- RepositoryMappingFilter interprets request URI as the path to a node
- AggregatorFilter analyses the request and stores the results in the [AggregationState](#) (Page and data Node, template e.t.c) → Node is loaded from JCR
- RenderingFilter renders page based on the data collected in AggregationState

## III. Proposals

---

### Duplicate current node as preview node

It's required to have ways (Context, query params) to determine which node/data need to be retrieve (current node or preview node). We can have **another filters to take over/overriding** the way we handling data e.g:

Alternative filter for RepositoryMappingFilter and produce the node path to preview node instead.

- Pros** data can be a draft data until it's synced to the current node. I recommend to leave the preview node there, hence the editing does not bind to specific session, People can return and continue to edit at any time.
- Cons**
  - Draft data must be sync based on actions: Switch tab (tab mode), editing (Split mode), select template, date e.t.c...

Draft implementation:

New content node is added this implementation. It's not enough and leads to having different errors and also for split mode there're properties editable we can't predict exactly what item/property will be changed. That's why I would go with duplicated node and will be hidden for enduser

[Main](#)

[Pages](#)

[UI Framework](#)

### Save working node in current session/context

Node will be saved to session whenever user enters/edits in the editor. When render the page, AggregatorFilter needs to be changed or overridden by other filter to decide where to load node JCR or Session. Again It's required to have ways (Context, query params) to determine which node/data needs to be retrieved

- Pros** Node data always be the same as editing node. No need to care about migration/maintenance of data and because of saving those stuff on session, all node data is only live within user session → swiped out once user logout or session is destroyed
- Cons** saving stuff in session is never be recommended, could lead to memory usages on server side and same as first option data must be sync based on actions. There's common thing in both solution 1&2 is data handling is not configurable in renderable definition.

Draft implementation: Mostly the same as the first one, but need to store working node to session instead of add new preview node

### Expose/Use Renderer Engine directly on tab UI

Since this solution is just an interoperation between tabs to produce needed out put. So CMS filters will be ignored when preview node is enabled

- Pros** Node data is loaded once and be interoperated within context of UI, no need to maintain the states of Node

- **Cons** the rendering processes is going throughout complicated processes/filter. We need a big effort to refactor those stuff and also maintenance of rendering in different places.

Draft implementation:

This draft implementation just use raw html string to provide component data, haven't touched the rendering yet.

[UI Framework](#)

[PAGES](#)

**Note:** all those options are required to update state/value whenever data change, either in the node or the cloned node.

### III. Discussion, Suggestions, Q&A

---

Feel free to comment or edit the page for any better idea.