Sling Resource Merger

The Sling Resource Merger provides services to access and merge resources. It provides diff (differencing) mechanisms for both:

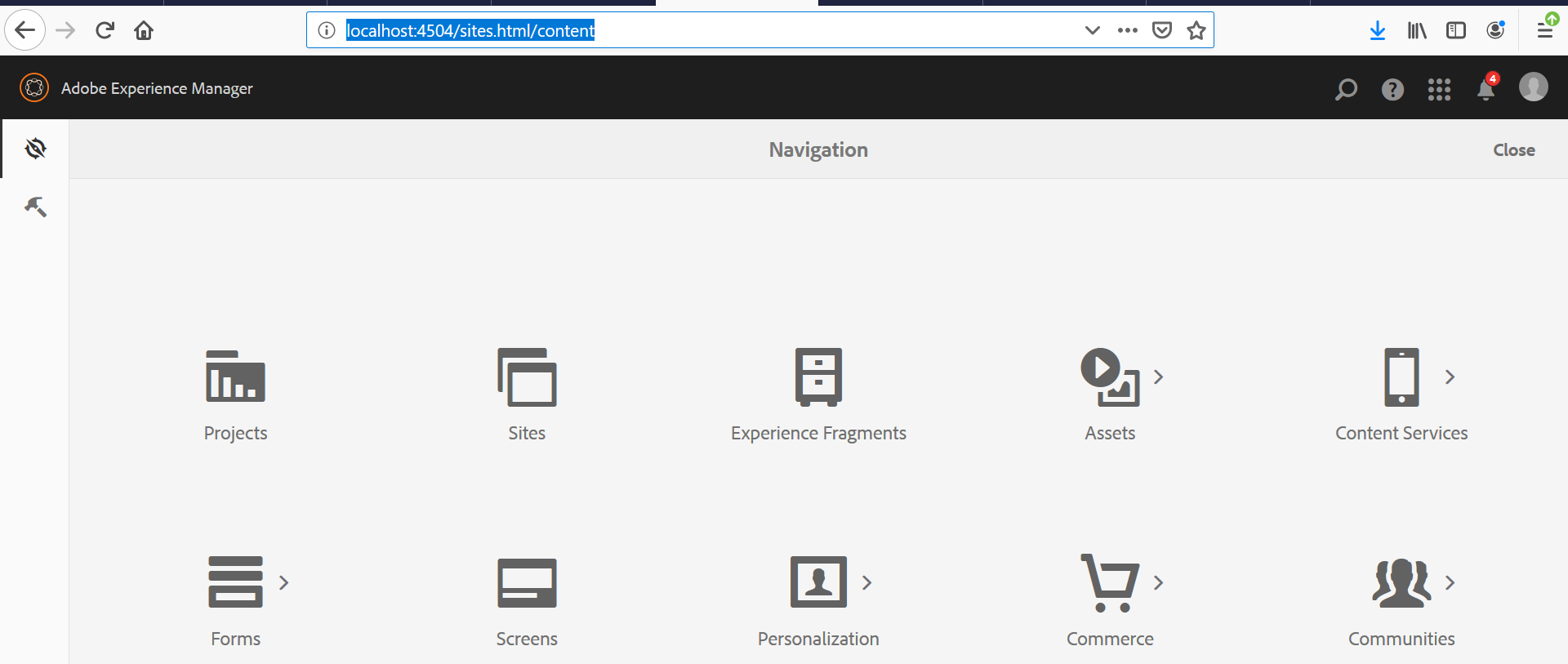
* **Overlays** of resources using configured search paths
* **Overrides** of component dialogs for the touch-optimized UI **(cq:dialog)**, using the resource type hierarchy (by means of the property sling:resourceSuperType).

1. **Approach for Overlaying Resources**: Resource Overlay works on the path which is having the same hierarchy in /apps. If there is a path in libs: **“/libs/text/example” then** its overlaying path can only be: **“/apps/text/example”.** This approach saves you from copying the entire structure of libs in apps when you need customizations. You can customize as much as required.
2. **Approach for Overriding Resources**: This approach works on the property **sling:resourceSuperType**. This concept is also called inheritance of the resource in AEM.

**Note**: **sling:resourceMerger** concept is basically used for Granite. So this concept is specifically applicable for touch UI.

# Overlays comes in case of customization

Ex: Adding a new item/console in the navigation section.



Steps:

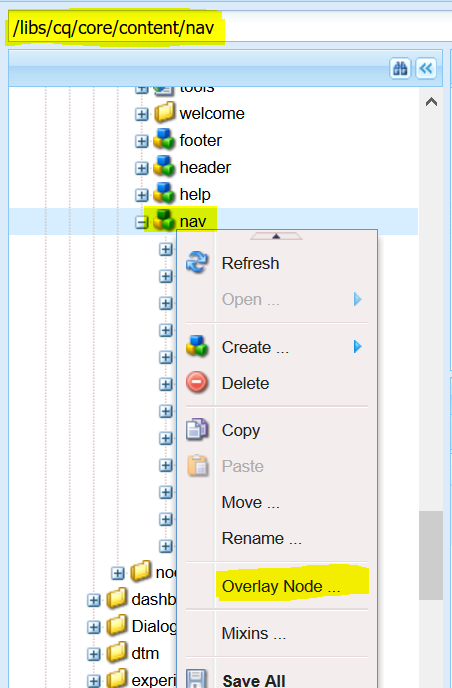
1. Go to the below mentioned path in crx de -> Right click on “nav” node -> Click on Overlay node.

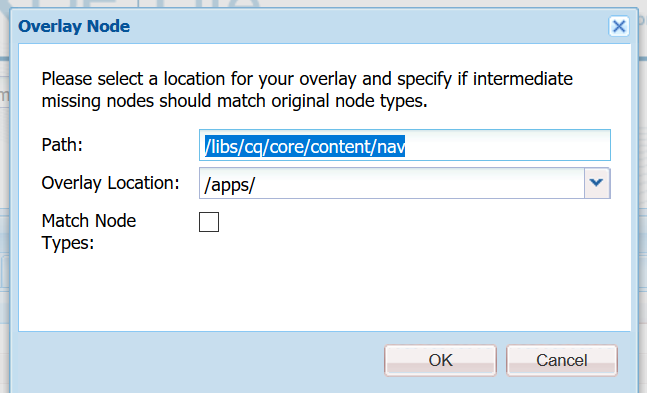
/libs/cq/core/content/nav

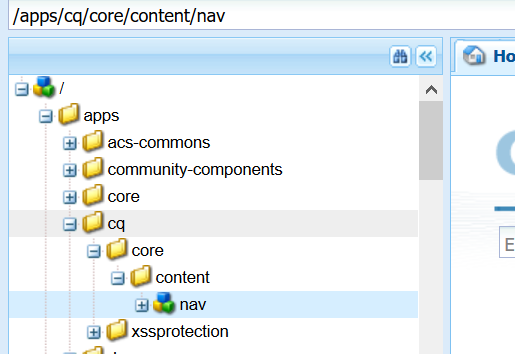
This action would show a popup, where path value would be “/libs/cq/core/content/nav

” and Overlay location would be “/apps/”

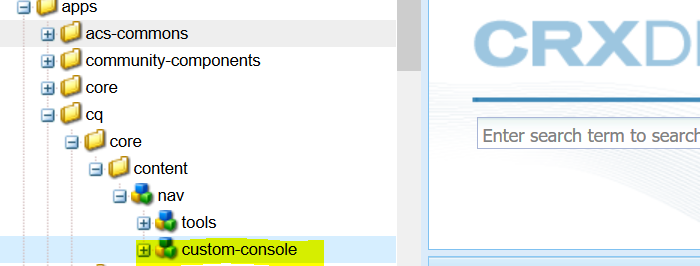
Clicking on “Ok” would create the node “nav” inside apps folder (/apps/cq/core/content).



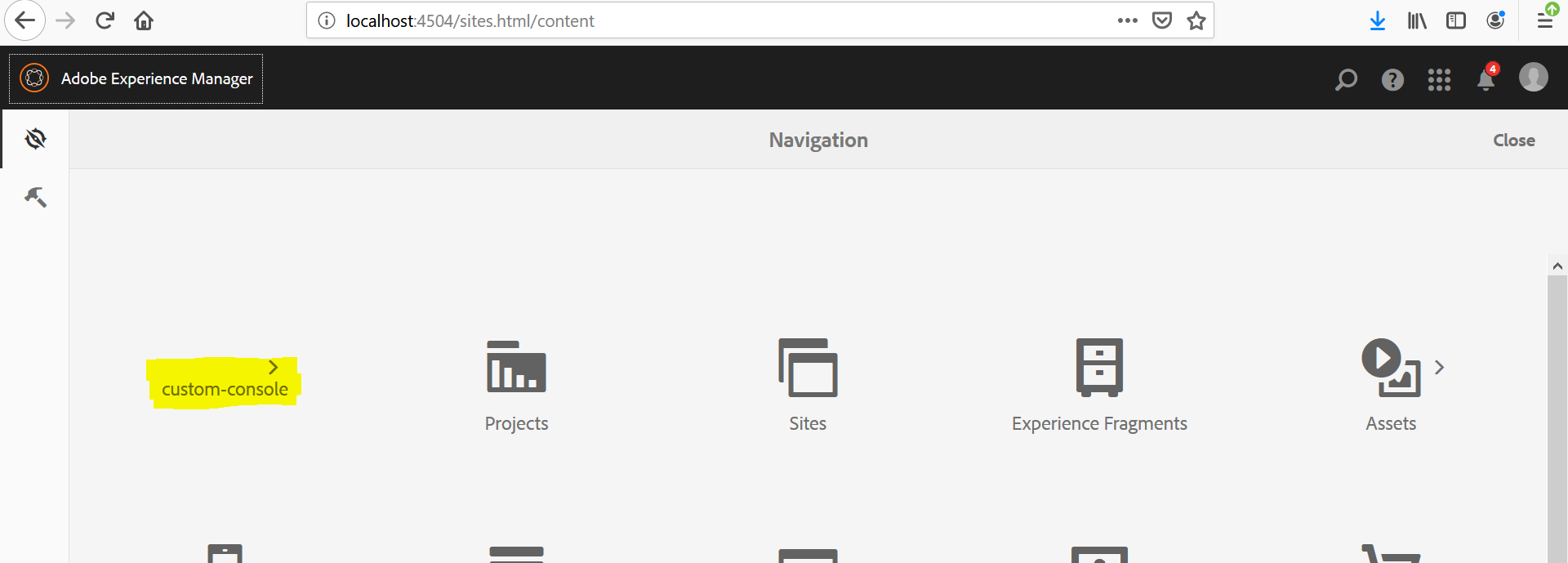




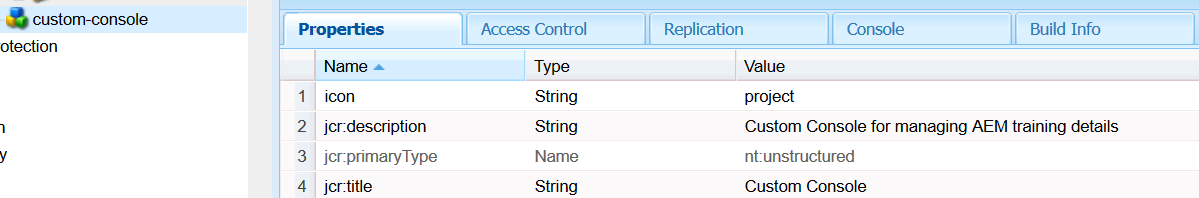
1. Right click on “nav” node which was created under apps folder in earlier step -> Create node of type “nt:unstructured”, with name as required (“custom-console” in my case).



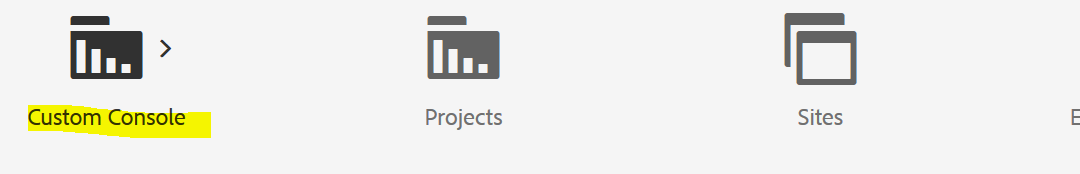
Result:



Adding more details, like title description and icon would give a better look and feel for the custom item.



Result:



In the above example, we don’t need to create the complete nav hierarchy.

**Modifying the existing items in navigation:**

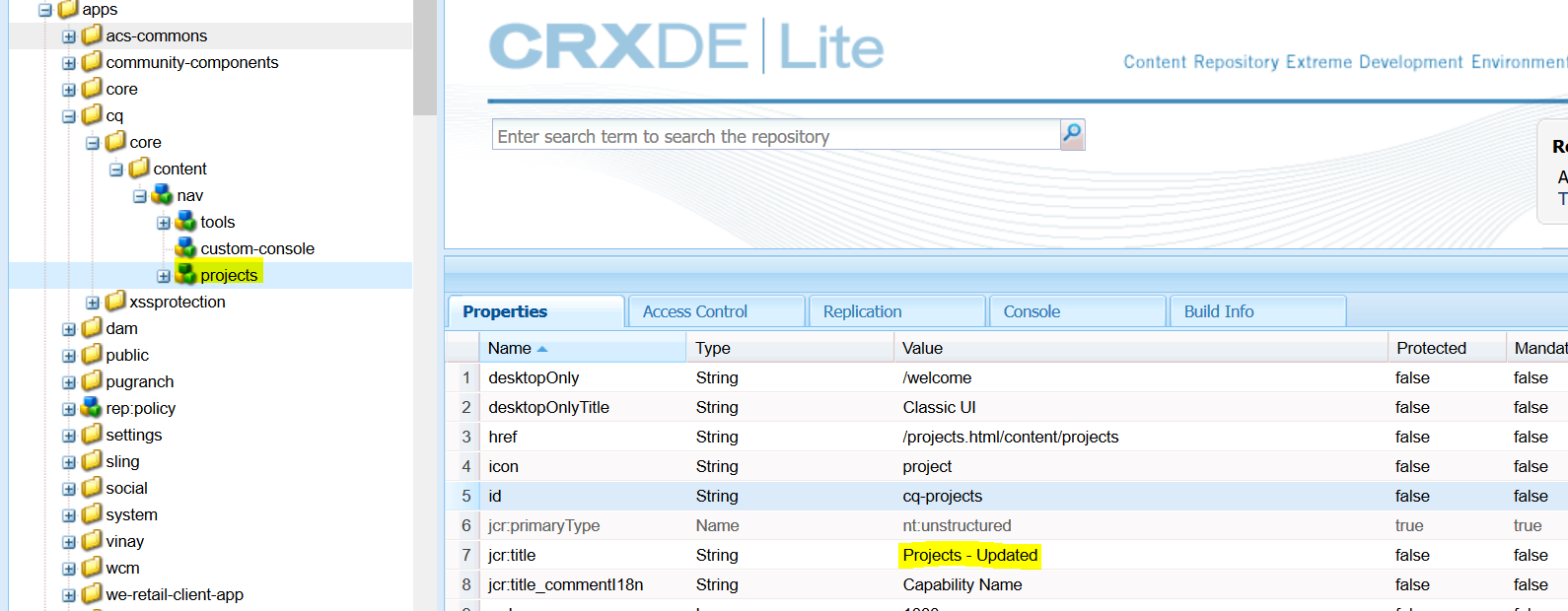
Steps:

1. Copy and paste the item that you want to modify from “/libs/cq/core/content/nav” to “/apps/cq/core/content/nav”.

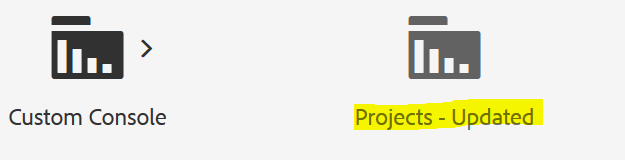
Ex: Copied projects node.

1. Modify the values that you wish for. It will reflect in the page console as shown in the below screen capture.

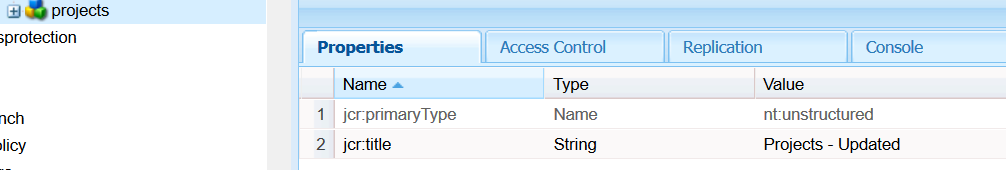
(I have updated the item title here)



Result:



**Note:** We can delete the remaining properties on the projects node, because those properties would be picked from libs location.



**Overriding concept:**

Overriding concept works based on the property called “sling:resourceSuperType”.

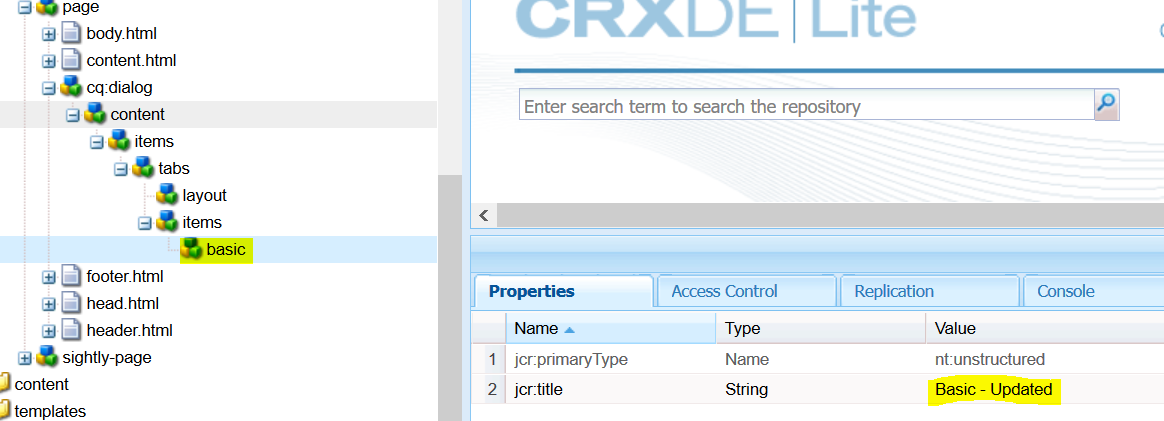
**Problem statement:** Add or modify the existing page properties dialog.

**Solution:**

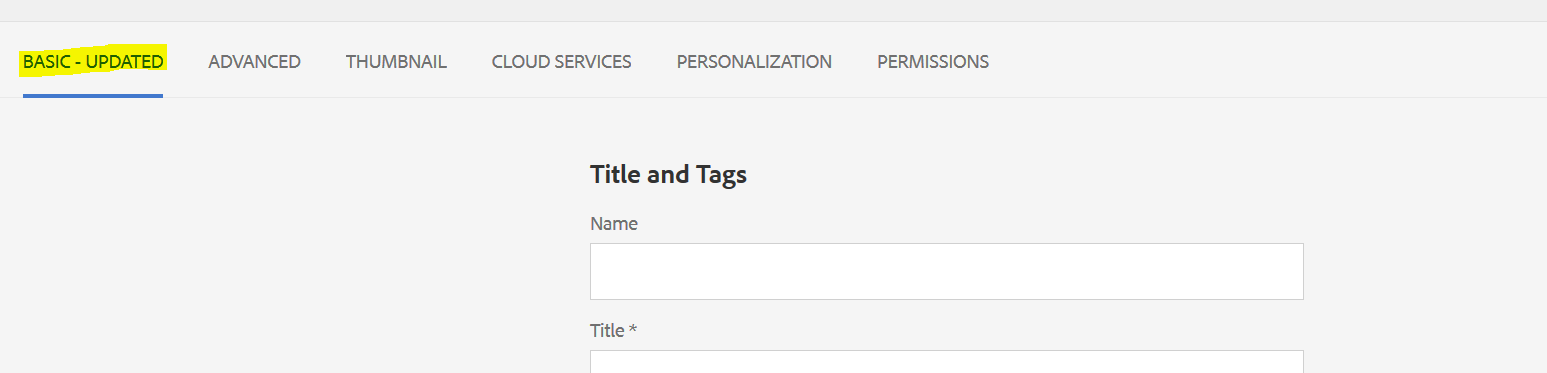
Copy and paste the cq:dialog node from the page component location (which is referred as the sling:resourceSuperType) to your page component.

Modify the existing property values to see the changes.

Ex: I modified the tab title.



**Result:**



**Note:** Though we don’t have any other properties and child items under “basic” node, it would still behave the same way as it was before. Because, sling will merge this basic node which is present under apps with the one that is present under libs.

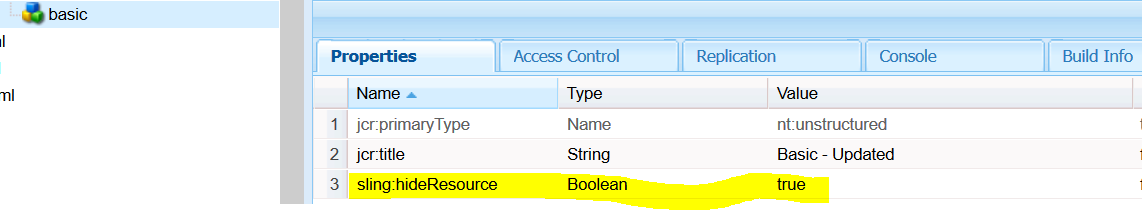
We can even remove all the properties from parent nodes of “basic” node.

**Hiding resource**

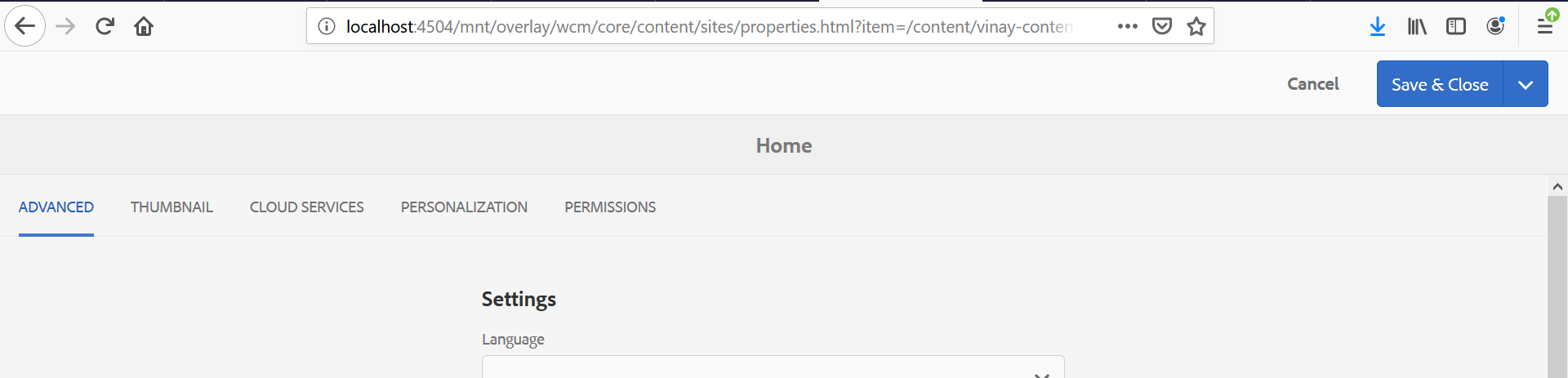
**Approach 1: Using “sling:hideResource”**

**Problem statement:** Hiding basic tab from page properties.

**Solution:** Add a property named “sling:hideResource” of type Boolean and value as “true”



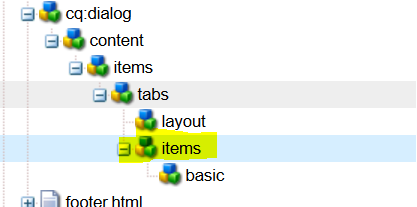
**Result:**



**Approach 2: Using “sling:hideChildren”**

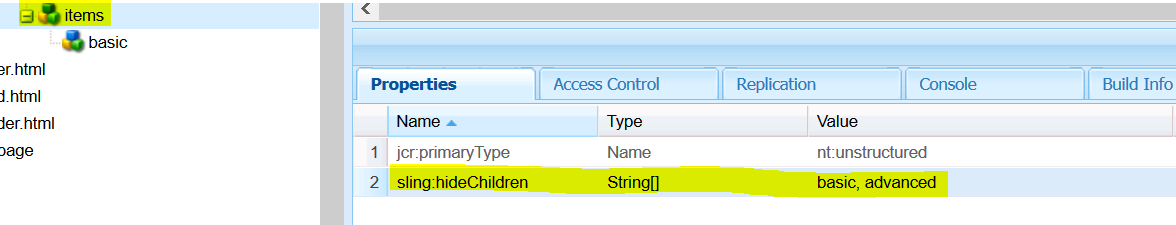
Steps:

1. Go to the parent node of the tab nodes.

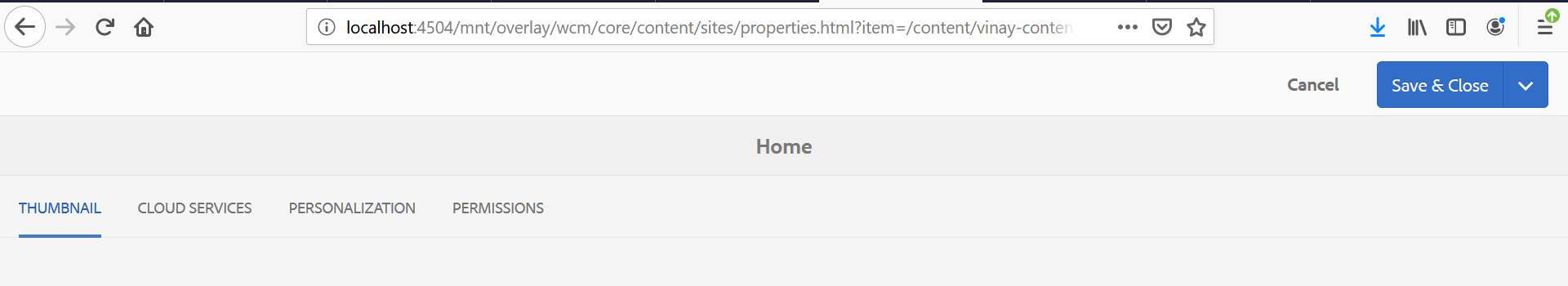


1. Add “sling:hideChildren” property of type string array. Provide the tab names which should be hidden from the page properties dialog.

Node: We don’t need to have the child nodes here under items.



**Result:**

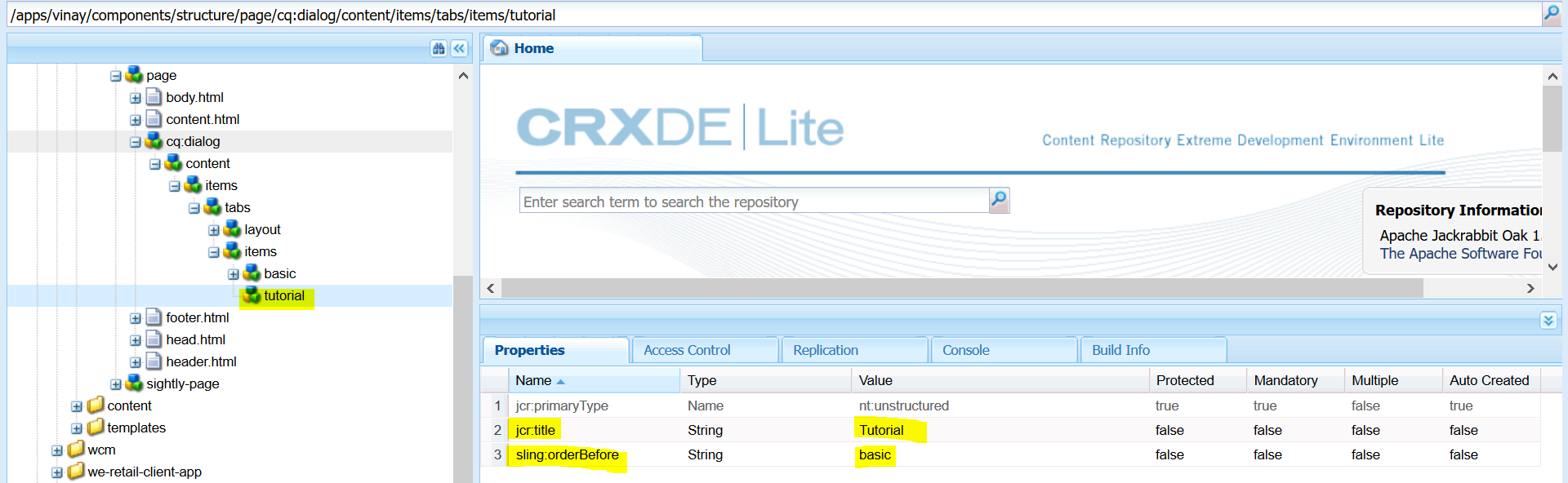


**Adding new tab in page properties and ordering it before the specified resource**

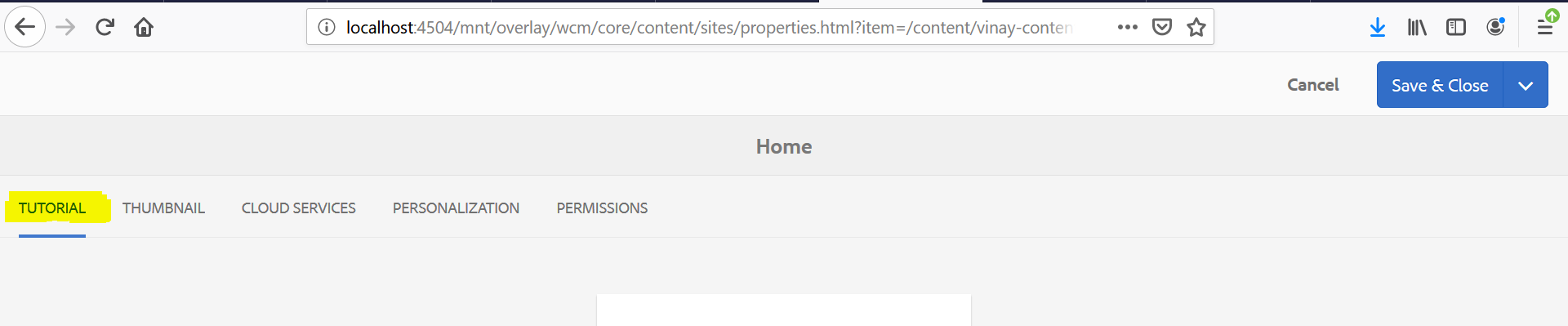
**Steps:**

1. Copy and paste the existing tab from cq:dialog.
2. Update the property values as required.

Ex: I have added “Tutorial” tab in my case.



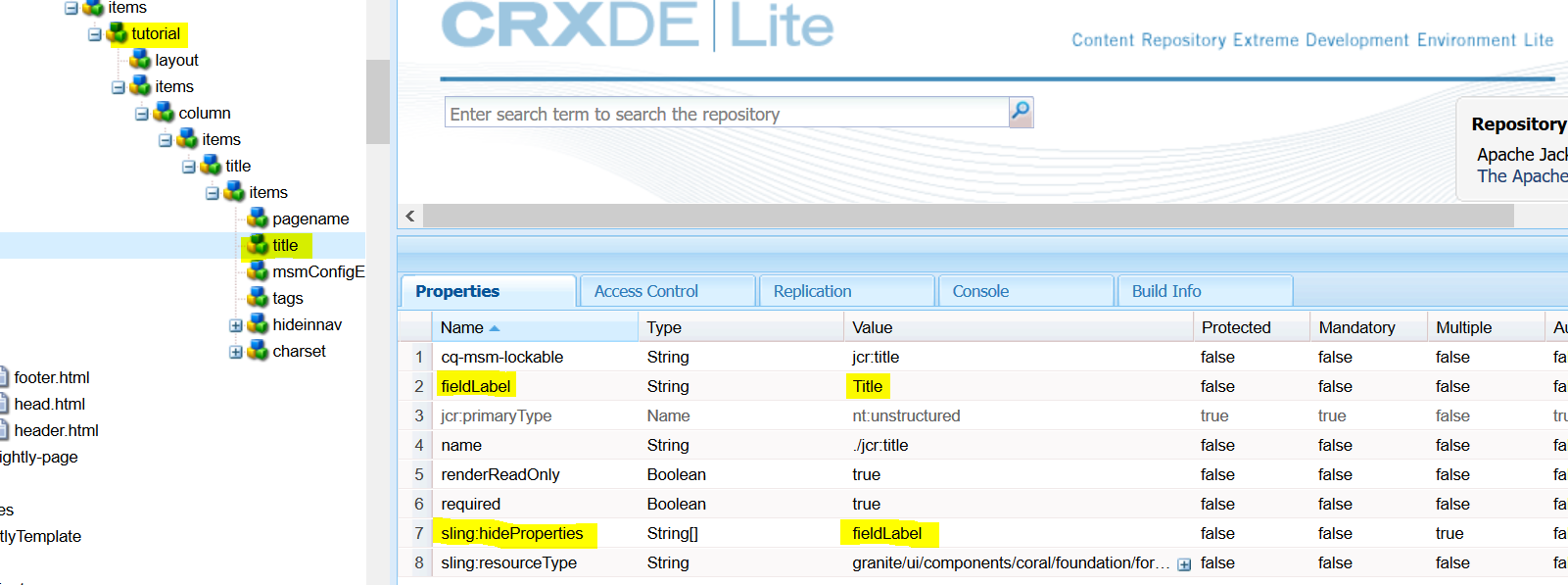
**Result:**



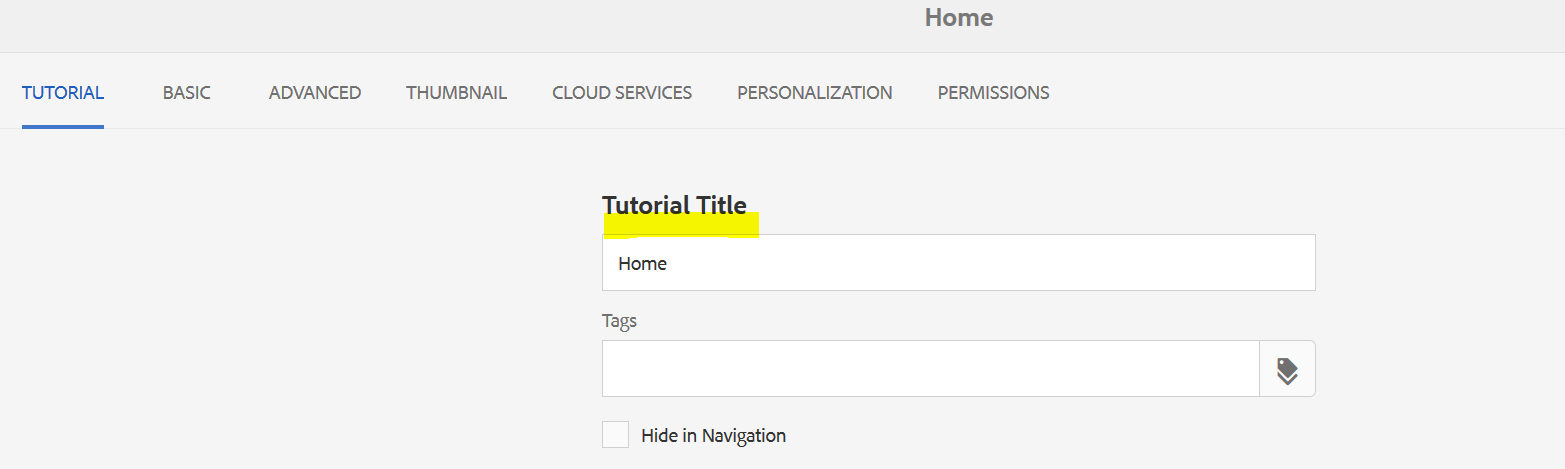
**Hiding field properties**

**Problem statement:** Hide field label from one of the dialog fields.

**Solution:** Add a property named “sling:hideProperties” of type string array and add the properties which you don’t want for that field.



**Result:**



Though there is fieldLabel value present on the node, as we have added “sling:hideProperties” property to hide the fieldLabel, it will not get displayed.