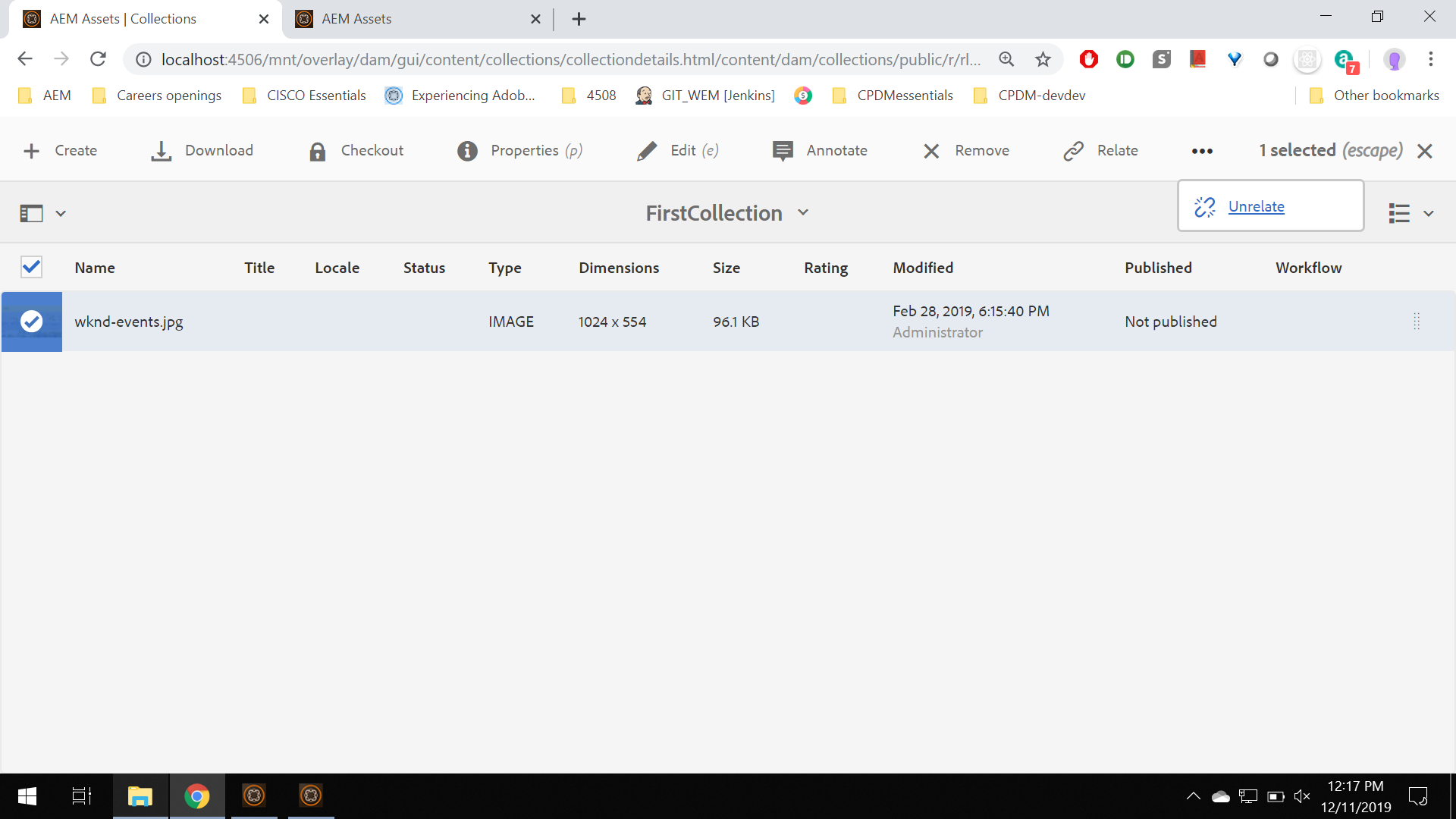
Using sling resource merger, we can

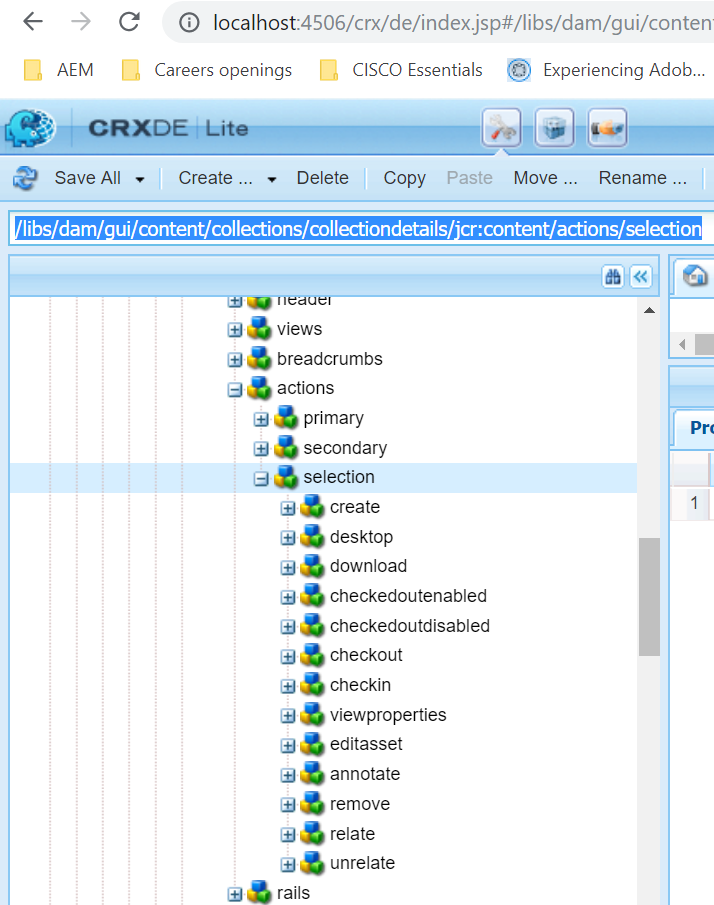
1. Create new buttons with corresponding actions
2. Edit existing buttons and their properties
3. Reduce on the no of nodes copied from /libs
4. Merge the resources available at ***/apps/a/b/c*** as ***/libs/a/b/c***+***/apps/a/b/c***

# Add a new button in collections console

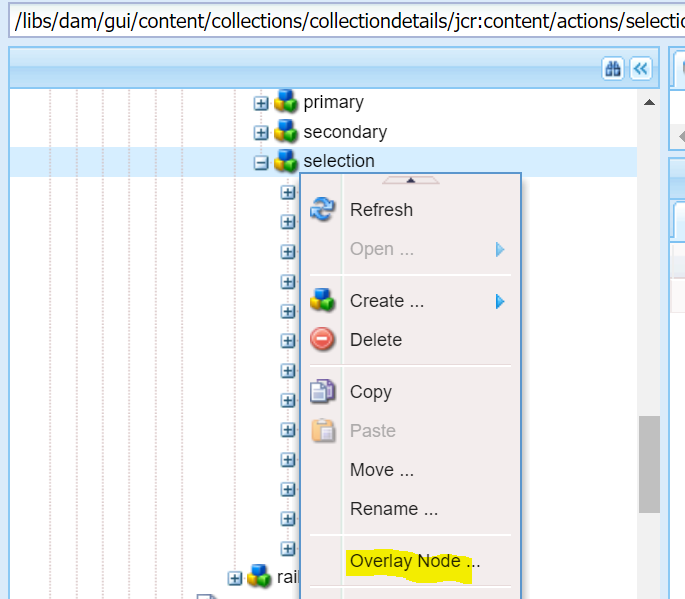
Right now below are the actions available



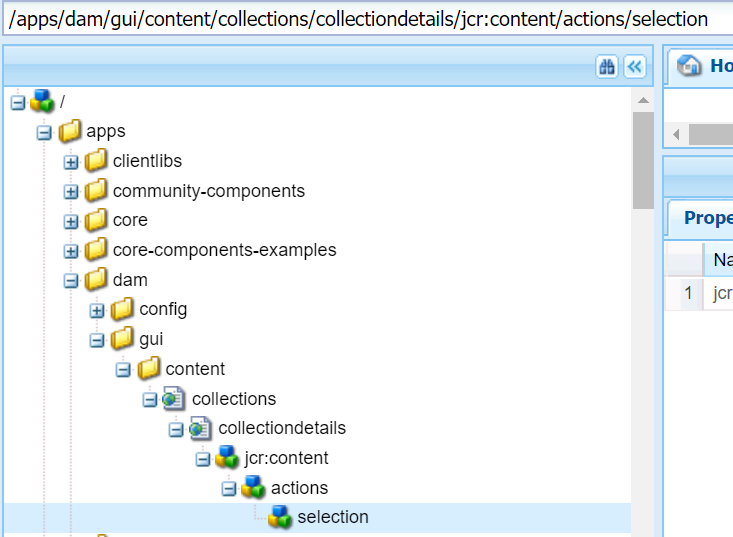
1. To add a new action button anywhere in the action bar, overlay the ***/libs/dam/gui/content/collections/collectiondetails/jcr:content/actions/selection*** to /apps



Right click on ***selection*** node and overlay as below

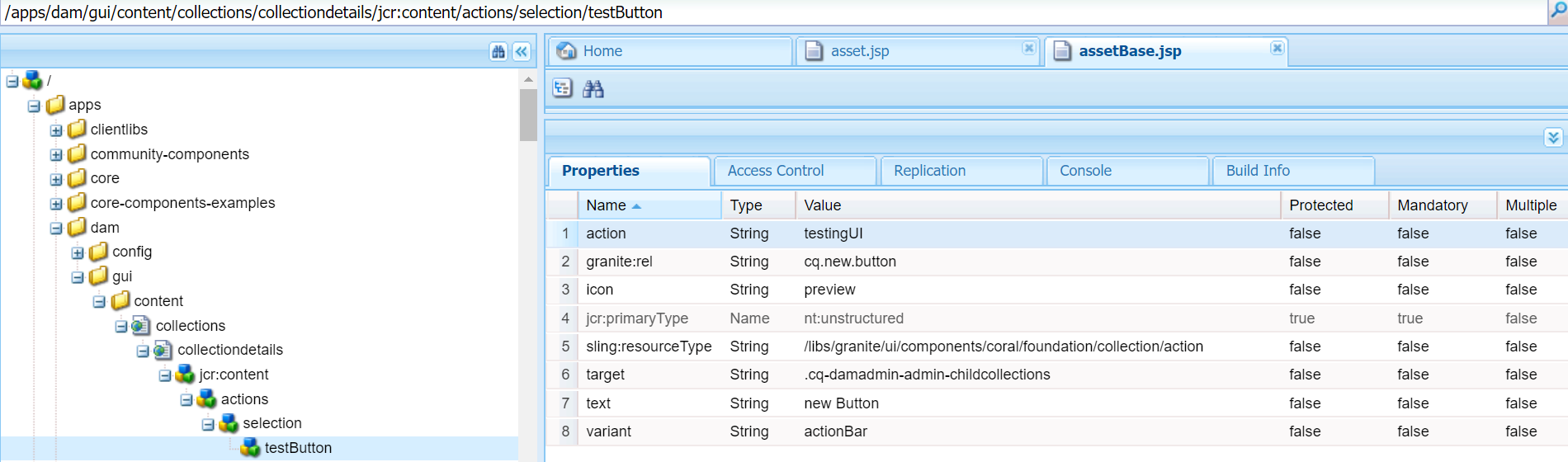


We get the similar node structure under ***/apps*** like below



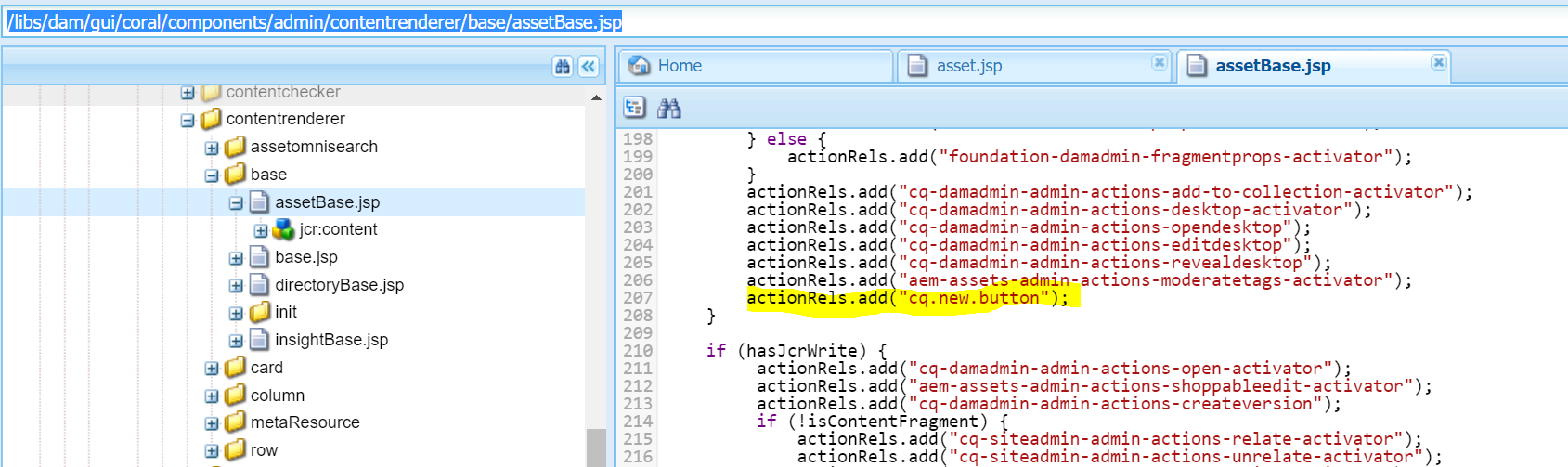
Note : All the coral UI OOTB things are available at ***‘/libs/granite/ui/components/coral/foundation’***

1. Here we can create a new node of type (nt:unstructured) and give some properties to it as below

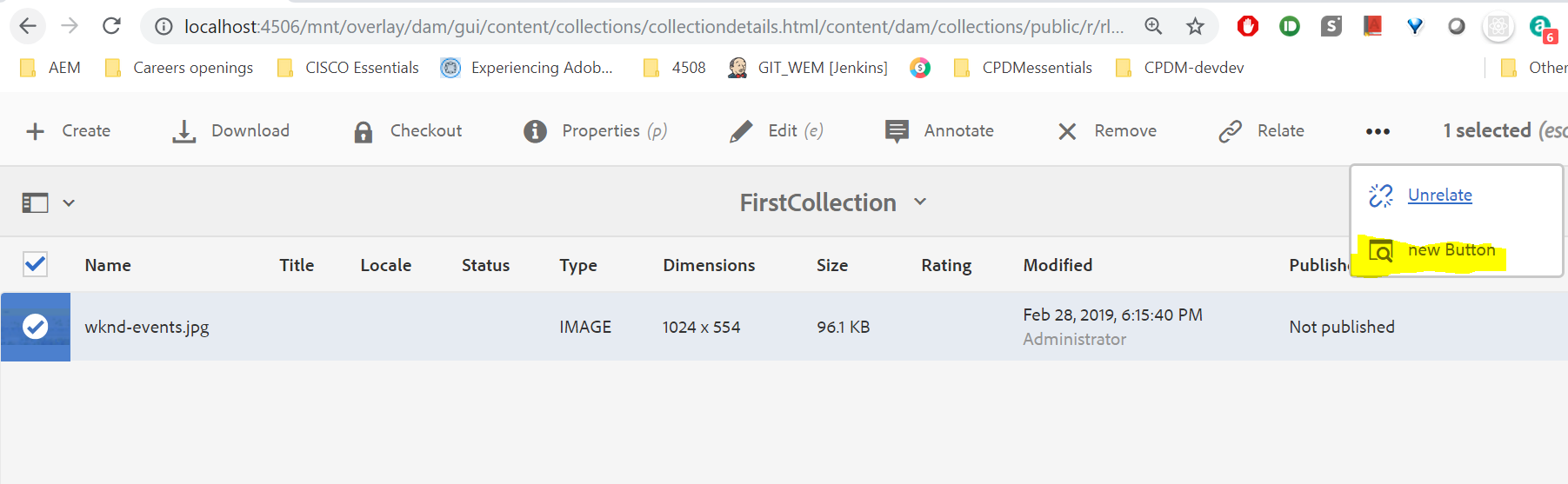


|  |  |
| --- | --- |
| Granite:rel | Adding a class to the button |
| Icon | To show any icon before the text of button |
| Sling:resourceType | Points to the granite UI action component |
| Target | Indicating the selections from a particular console will be acted up on. This property value can be found at the <title> in elements tab of the AEM console. This property is optional |
| Text | Text to be visible on the button |
| Variant | This indicates what type of element are we adding to load stylings based on that |

1. Now we are done with creating a new button in CRX. For this to be visible in UI, we have to add this in ***/libs/dam/gui/coral/components/admin/contentrenderer/base/assetBase.jsp.*** This Jsp is responsible for fetching all the buttons depending on the access of logged in user.



1. Post this the created button comes up in UI as below



1. Now to configure an action to the button in UI, we will have to register the action using granite UI API as below

$(window).adaptTo("foundation-registry").register("foundation.collection.action.action",{

name : "testingUI",

handler : **function**(name, el, config, collection,

selections) {… this is where all the JS code goes..}

});

1. This js file has to be group under a clientLibrary folder and some categories assigned to it.
2. The above clientLibs have to added to dependencies of ***head***  at ***/apps/dam/gui/content/collections/collectiondetails/jcr:content/head/clientlibs(overlaid node).***

# Useful Links

C:\projects\dragon\_release\jcr\code\jcr\_root\apps\cq\gui\components\common\wcm\chp\_publish\js\chp\_publish.js

C:\projects\dragon\_release\jcr\code\jcr\_root\apps\cq\gui\components\common\wcm\previewRendition\js\previewRendition.js

/apps/dam/gui/content/collections/collectiondetails/jcr:content/actions/selection/pagepreview