

Concept - Delete Action

- Abstract
 - Requirements
 - Delete a Content
 - Action on a Deleted Content on the Author instance
 - View Previous Version
 - Restore previous version
 - Start the workflow process to validate the changes
 - On the Public instance
- Proposal (for pages app)
 - Delete Action
 - Show previous version Action (markAsDeleted)
 - Restore previous version Action (markAsDeleted)
 - Activation of a Node Marked as deleted (markAsDeleted)
 - Pages Action for a Node Marked as deleted (markAsDeleted)
- Current Issues
 - Display Deleted Icon on the Content view (Tree/List)
 - Close subApp
 - Double click handling and default action.
 - Display Specific action for Node mark as deleted
 - Version Handling (ok, solved).

Abstract

Create a mark as delete Command for publishable content.

Related User Story.

-  Unable to locate Jira server for this macro. It may be due to Application Link configuration.
-  Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Requirements

Delete a Content

If the delete action is linked to a Content workspace (workspace containing content that can be published), delete Command will have a Confirmation dialog:

- Request a confirmation from the user.
- Display additional information's if the page/content is related/linked to others contents (*only for ee*)



Action on a Deleted Content on the Author instance

Once the user validate the deletion, the node and sub-nodes are marked as Deleted (Red cross Dialog).

[View Previous Version](#)

Open the Previous version (last valid version) in preview mode.

[Restore previous version](#)

Restore the Previous version (un-mark the current page as deleted)

[Start the workflow process to validate the changes](#)

Send a workflow request for this modification. Once the request is accepted, the current page is deleted.

[On the Public instance](#)

The **content is still available** till the Deletion is validated on the workflow side.

Proposal (for pages app)

Delete Action

- Create a confirmation dialog displaying the following text:
Delete node?
The node
/demo-features
and its sub nodes will be marked for deletion.

The following specific message will not be displayed as **DependenciesDialogControl/DependenciesServlet** used to define these dependency is a EE feature (module-content-dependency).

They will be deleted from all instances, once you activate this change.

Warning: this node will be marked for deletion (you will be still able to undo it) but dependencies to it have been found. If you will confirm deletion, the following pages will have broken references. Are you sure you want to continue?

/demo-project

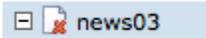
/demo-project/about

/demo-project/about/history

/demo-project/about/subsection-articles

/demo-project/about/subsection-articles/large-article

Content view should be refreshed based on the executed command:

- markAsDeletedCommand
 - Change the Node Icon Style

 - Keep the focus to this node.
 - Change the Action Bar available Actions:
 - Activate Changes
 - Show previous version (Version before deletion)
 - Restore previous version

Show previous version Action (markAsDeleted)

Display the last version of the Page marked as deleted in a **preview format**.

Create a Version Action

- Based on the VersionManager, get the previous version (a node 1.0 mark as deleted has a version 1.1)
- Call the pageEditor with this version.

Adaptation

- Default location has to be able to handle the versions.
- Page editor has to retrieve the correct Page version in case of version is present

Restore previous version Action (markAsDeleted)

Re-publish the node mark as deleted.

Create an RestoreVersionAction/Definition.

Activation of a Node Marked as deleted (markAsDeleted)

Pages Action for a Node Marked as deleted (markAsDeleted)

When double click on a deleted page, create a new Editor sub app with a dummy page 'PAGE MARKED AS DELETED... ' . Available actions displayed in the action bar should be the same as in Show previous version Action (markAsDeleted)

Current Issues

Display Deleted Icon on the Content view (Tree/List)

Node marked as deleted are of type MixinNode (NodeTypes.Deleted).

Defining a workbench/nodeType in the configuration, this has no effect as in ContentPresenter.getItemIcon(..),

JcrNodeAdapter.getPrimaryNodeType() return the primaryNodeType (for a mgnl pages that is marked as deleted, this always return mgnl page).

Solution

Add a generic feature in ContentPresenter that always check for the Deleted Node. In case of Deleted Node, Display a common Icon.

Close subApp

Use case:

Open the deleted node (preview last version) --> open the sub app in preview mode.

When clicking the activate (activate deletion) --> call the activate Action.

- Activate action send a ContentChangeEvent to the AdmincentralEventBus
- PageEditorPresenter listen to this Event but on the SubAppEventBus

How to notify the PageEditorPresenter that an action was performed and in case of node do not exist--> close.

Solution:

PageEditorPresenter will register a listener to the AdmincentralEventBus.

Show a Warn message before closing the view 'Item do not exist anymore'

Double click handling and default action.

Double click on PagesMainSubApp is handle by the BrowserPresenter.executeDefaultAction() --> defaultAction = subAppDescriptor.getActionbar().getDefaultAction();

How to solve?

Solution:

No more an issue as we reintroduce the Default Deleted Template.

Display Specific action for Node mark as deleted

Ouf, pages app is not yet migrated to the Restriction definition. Is the restriction able to handle mixin Node type?

Version Handling (ok, solved).

We have to create a previewLastVersion action. Issue is that Location don't support Version.

<http://localhost:8080/magnolia-bundled-webapp/demo-project/about.html?mgnlVersion=1.0> --> ok

<http://localhost:8080/magnolia-bundled-webapp/.magnolia/admincentral?restartApplication#app:pages:detail;/demo-project/about?mgnlVersion=1.0:edit> --> ko

Currently node is retrieved in PagesEditorSubApp calling

```
path = location.getNodePath();
```

```
Node node = MgnlContext.getJCRSession(workspace).getNode(path);
```

```
path is /demo-project/about?mgnlVersion=1.0....
```

Solution:

Extends the DetailLocation in order to be able to pass as parameter the version.
Adapt the parsers.
Expose a getVersion and has version().

Pass the version as parameter to the location. In case location contains a Node version, retrieve the specific version.