Comala Metadata app

Overview

This page details integration between Comala Metadata app and Comala Document Management.

You will need **Comala Metadata app version 3.4 or above**.

The Comala Metadata app was the basis for the Comala Document Management own metadata functionality, and as such both apps are innately compatible with each other.

If you are using metadata values created with the Comala Document Management **{set-metadata}** macro, these are only indexed by Confluence and appear in the Confluence Search if the Comala Metadata app is installed.

> If you have trouble accessing the links to the Comala Metadata app documentation on this page, try another browser (Firefox, Safari, etc)

Accessing Metadata Properties

Use **Value References** to the metadata field you wish to retrieve.

For example, to get the value of a metadata "Author" field, use:

```
@Author@
```

If you want to retrieve metadata from another page, you can include the page title:

```
@Page Title:Author@
```

And you can retrieve values from pages in different spaces by including the Space Key:

```
@KEY:Page Title:Author@
```

For more information on using these, see: Value References and Metadata references.

Accessing Additional Internal Metadata

You can also reference additional internal metadata.

Accessing Metadata Supplier values

Values from the Metadata Supplier can be accessed via Supplier References.

Saving to Metadata

Use the Comala Document Management **`{set-metadata}`** macro to store metadata properties which can then be read by the Comala Metadata app **`{metadata-from}`** macro.

The **`{metadata-from}`** macro can also read page-level workflow parameters when the **`{workflowparameter}`** macro is used with the **`edit=true`** parameter.

Accessing Metadata from User Macros

You can also access metadata from Confluence User Macros, and access to it from your custom workflows.

For example, to set an *auditor* in a user macro, as a metadata value, you need to set it as *metadata.auditor*:

```
#set ($containerManagerClass=$content.class.forName('com.atlassian.spring.container.ContainerManager'))
#set ($getInstanceMethod=$containerManagerClass.getDeclaredMethod('getInstance',null))
#set ($containerManager=$getInstanceMethod.invoke(null,null))
#set ($containerContext=$containerManager.containerContext)
#set ($contentPropertyManager=$containerContext.getComponent('contentPropertyManager'))
##Set property
$contentPropertyManager.setTextProperty($content, "metadata.auditor", VALUE)
```

You can validate that the value is properly populated, again with a user macro:

```
##Test property
#set ($managerProperty = $contentPropertyManager.getTextProperty($content, "metadata.auditor"))
<p>$managerProperty</p
```

3.- Finally, we reference that metadata with a value reference *@auditor@*:

```
{workflow:name=Metadata example}
  {state:Review|approved=Published|rejected=Review}
    {approval:Editor in Chief|user=@auditor@}
  {state}
  {state:Published|final=true|updated=Draft}
  {state}
{workflow}
```

See also

- Product Documentation
- Metadata references
- Supplier References