

# Link API improvements

- Current use cases:
  - In CK editor, when defining a link to an image

## Current use cases:

### In CK editor, when defining a link to an image

The following is currently generated `${link: {uuid: {0fa615af-207d-4a45-a3aa-9103d76e54d9}, repository: {dam}, handle: {/demo-project/about/subsection-articles/article/content/00/text_files/file}, nodeData: {resource}, extension: {jpg}}}`

This uuid refers to a dam node identifier but should refer to an asset composite key such as `jcr:0fa615af-207d-4a45-a3aa-9103d76e54d9`

To achieve this, we need to look into 2 areas:

- CKEditor and our plugin - where we generate such links
- The link API - where we parse it

**The current self-imposed limitation is that assets selected in CKEditor will link to the original asset, not to a variation.** Therefore, it shouldn't be too much of a problem to stash the DAM composite key in either of the uuid or handle fields above (although it does seem a bit hackish).

The proposal below is left for reference but is more than what we can handle for 5.3

---

```

${dam-link: {key: {jcr:0fa615af-207d-4a45-a3aa-9103d76e54d9}}}
    since we're in a specific link type, we're free to use whatever attributes we want. Not sure any of
the other attributes would make sense in the case of DAM.
    we introduce different formats per link type (alternatively we keep ${link}, default to the Website links, and
add a type attribute)

# in the case of dam links, I see two possible extra attributes:
# inline: true/false - a simpler notation for Content-Disposition, somehow passed to the download servlet
# data: true/false - would generate a regular URI (false), or a data URI (true)
# come to think of it, these could be a single attribute with several possible values
# perhaps we could keep the extension, as an indicator of the type of link? Or added a mime-type attribute.
# perhaps we could generally add an alt-text attribute (although ${link} is really meant for the hfr
```