Examples of Using the "Fluid Wiki Resources" Macro

Goal Statement: while I don't want to recreate Zotero in Confluence, I'm hoping we can have a method of entering the resources that has no more overhead than entering and formatting them as standard Confluence links (and might even save time), while also ensuring the wiki page output has a machine-readable structure for future usage. A macro will hopefully provide us with greater (or at least equivalent) ease of use to simply entering links while providing some future-proofing against anticipated future desires.

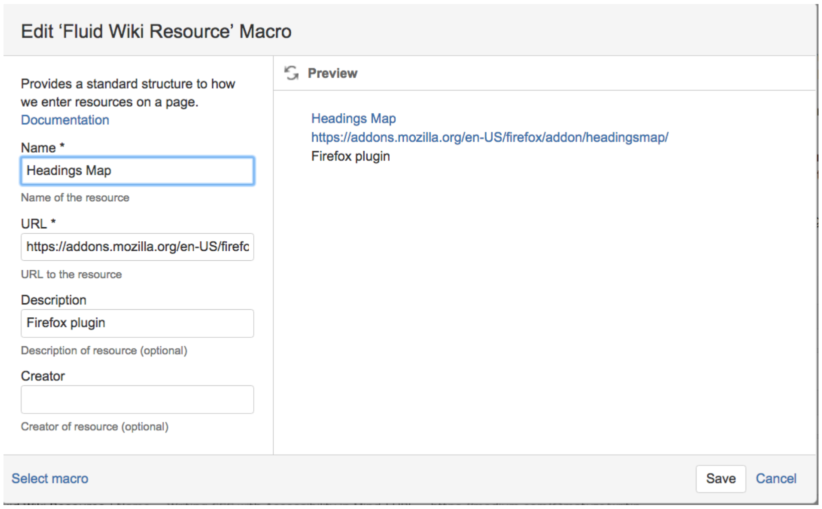
I (Alan Harnum) have experimentally made a custom macro (**Fluid Wiki Resource**) for this Confluence wiki that can be used to enter resources.

I believe there's two main advantages to using a macro for this:

1. Helps ensure we have consistent information captured and displayed about resources (we can have both required and optional fields, and write templates for outputting the information for each resource; if we update the template, the markup of each outputted resource will automatically update)
2. We'll more easily be able to parse and extract resources from the wiki using automated tools (this will help with getting them into Zotero or writing code to automatically generate resource lists in other contexts)

The easiest way to use the macro when entering content on a page is to type a **{** (left curly brace), which in the standard Confluence editor will open a "Macro suggestions" dropdown - you can then continue typing **{reso** or similar to find and insert the macro. This will display a similar interface to the one used to create a standard link, but with some further optional fields available (creator and description).

There are examples of using the macro below (and a screenshot). Please provide feedback to Alan Harnum on any aspect of this - the output template is very basic right now but can be easily customized to output different markup for each resource, or we can have other optional fields, etc.



Edit 'Fluid Wiki Resource' Macro

Provides a standard structure to how we enter resources on a page.
Documentation

Name *
Headings Map
Name of the resource

URL *
https://addons.mozilla.org/en-US/firefc
URL to the resource

Description
Firefox plugin
Description of resource (optional)

Creator

Creator of resource (optional)

Preview

Headings Map
https://addons.mozilla.org/en-US/firefox/addon/headingsmap/
Firefox plugin

Example of Harvesting Resources

Here is a simple example using RunKit of harvesting resources entered using the macro from the page and converting them into JSON: https://runkit.com/waharnum/5a70dd9238fd120014773566

Testing Tools

Headings Map
https://addons.mozilla.org/en-US/firefox/addon/headingsmap/
Firefox plugin

Readability Score
https://readable.io/

Some Useful Articles and Essays and Books

Writing CSS with Accessibility in Mind (Manuel Matuzovic)
https://medium.com/@matuzo/writing-css-with-accessibility-in-mind-8514a0007939
Tags: css, a11y

Nothing About Us Without Us (James I. Charlton)
https://www.goodreads.com/book/show/1429478.Nothing_About_Us_Without_Us