COMMENTS UX: PROVIDE A UNIFIED LINK INSERTION BUTTON AND SYNTAX WITH AUTOCOMPLETION, INSTEAD OF DISTINGUISHING BETWEEN INTERNAL (WIKILINK) VS EXTERNAL HYPERLINK TOOLBAR ACTION BUTTONS

Status

● Open

Priority

4 Good thing

Task

Comments UX: provide a unified link insertion button and syntax with autocompletion, instead of distinguishing between Internal (WikiLink) vs external hyperlink toolbar action buttons

Add a timesheet

Add a timesheet for this task

Description

This distinction is artificial and never made sense to me. Why have two buttons for hyperlinks, and why should users care / know the subtle difference between the two? Just provide one link button, and if you really must use different syntax between the two (I don't know why), then auto-figure-it-out at insertion time based on what I have put in the form.

Clicking the link button should let me type words with live search autocompletion (for internal page/items matches) like WordPress and Google Docs do, and/or let me type/paste a URL (and it should automatically figure out the title/metadata for me), whether internal or external (default to target="_blank" for external links) and just automagically handle it. Heck, even just pasting a raw URL into the contents should do that, live in the editing canvas. If typing an external URL (instead of pasting it) you will have to use my timeout-based heuristic to avoid spamming the remote server with opengraph requests.

I would suggest unifying the syntax there, unless you have some *really* good reasons not to; I posit the user doesn't care about the distinction between internal and external. Why should they?

Emails

Assigned to
developer

Assigned by
jeff.fortin-tam

Keep informed
jonny.bradley, lukas.masek, victor.emanouilov

Also visible to
1- Visible to Anonymous (Public content)

Related

- ● 4 Good thing: Search usability (auto-complete): Ensure good live search performance by using a minimum amount of characters and a more relaxed search activation timeout delay to avoiding hammering your servers assigned to developer
- ◑ 5 Nice to have: Rich text comments toolbar should have a few more essential buttons to encourage good writing form assigned to developer

Created
Friday 26 of August, 2022 14:35:56 EDT
by Jeff Fortin T.

LastModif
Friday 07 of June, 2024 14:03:53 EDT

Comments

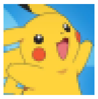
Wiki links are special

Jonny Bradley 2024-02-19 12:31

Wiki links get updated if you rename the page, for instance, so the syntax for them needs to be different, e.g. `((pagename))` not `[other]` so the user needs to choose at some point, we can't guess in advance because you can have a relative link on a wiki page that isn't to another wiki page e.g. login.

Admittedly if it starts with `https://` it will be an external link, probably... so we could combine the two dialogs and add a dropdown selector to choose, but this would make things tricky with ckeditor as it has its own external link dialog, same i think for Markdown and ToastUI.

Perhaps when all the other issues in tiki are fixed we could look at this, but it will be complicated (as most things in Tiki tend to be 😳)



The UI ought to make it easy to work with all link types without memorizing heterogenous syntax
Jeff Fortin T. 2024-06-07 14:22

Yeah, the fact that

- I have to remember the syntax for not only internal (double parenthesis?), external (mixture of something), but _also_ Tracker item IDs with square brackets, and;
- the comments system doesn't offer toolbar actions for links other than external hyperlinks, and;
- if I rememeber correctly, the preview tab in some cases (the "New tracker item" description field, for example?) doesn't render some link types even if I presumably use the correct syntax

...is overall quite counterproductive.
In Tiki, I don't have an easy way to link to other tracker items in a comment (or description) in shorthand form (without all the cruft from after the item number, see item81936 ...), unless I keep a cheatsheet next to me.

Maybe if you absolutely, positively *have* to have different syntax for different links type throughout, then it becomes just a matter of "*All* the UIs need to provide *one* unified 'smart' link insertion GUI with autocompletion, where the system parses/validates what the user meant and then inserts the hardcoded syntax for the correct link type."

However, it would feel so much nicer if there was no divergent syntax needed; as an example, GitLab (and to an extent, GitHub and Bugzilla) manage to parse anything you throw at them correctly, whether it is a link to an issue or MR internal to a project, or in another project within the same or different scope in the same server instance, or an external link. It just works.