

Bugs & Wish list

PluginTogether should replace the url associated with the Edit page button so that co-editors are automagically offered to go to the session with collaborative edition instead of the warning of edition conflict | Tiki Wiki CMS Groupware :: Development

PluginTogether should replace the url associated with the Edit page button so that co-editors are automagically offered to go to the session with collaborative edition instead of the warning of edition conflict

Status

⊗ Closed

Subject

PluginTogether should replace the url associated with the Edit page button so that co-editors are automagically offered to go to the session with collaborative edition instead of the warning of edition conflict

Version

18.x

21.x

22.x

Category

- Usability
- Feature request
- Community projects
- Dogfood on a *.tiki.org site
- Less than 30-minutes fix

Feature

Edit interface (UI)

Resolution status

Fixed or Solved

Submitted by

Xavier de Pedro

Keep informed

Victor Emanouilov, Marc Laporte

Lastmod by

Xavier de Pedro

Rating

★★★★★★★☆☆★★★★★☆☆★★★★★☆☆★★★★★☆☆★★★★★☆☆★★★★★☆☆ (1) ?

Related-to

- ⊗ PluginTogether fails due to warning of edition conflict

Description

See first the box saying "Even better" below.

"PluginTogether should replace the url associated with the Edit page button so that co-editors are automagically offered to go to the session with collaborative edition instead of the warning of edition conflict" (was "~~Clicking on PluginTogether on a wiki page should provide a CoEdit button next to Edit with the right url for all.~~")

This would be useful for taking meeting minutes collaboratively in the TRM and TAG meetings, both

which take precious time from some active tiki community members but with little time to do things properly (wasting precious time in this note-taking task instead of being able to invest it in other areas requiring their help too).

I've been trying to use tiki for many semester with my students at uni but I can't get them to use it with realtime collaboration with ease, so that they move to googledocs.

This minor usability change would be probably enough to convince teams to use tiki realtime collaboration feature with plugin togetherjs.

Like ourselves when taking notes on a roundtable meeting, or tiki admin group meeting, etc.

Even better

Even easier for the end user would be to just offer a user that sees the concurrent edition warning box, a link to carry on and convert the session into a togetherJS session in which both users continue writing together. In the backend, tiki would probably:

1. A new general preference should be added for this new behavior to be enabled
2. when user A clicks at edit a wiki pageFoo, the url that would be created by PluginTogether is stored in tiki mysql tables somewhere, with a semaphore/flag or similar so that tiki knows that this page has been started edition and the url to allows others to go into the same edition session collaboratively.
3. when user B clicks at the Edit button of the same pageFoo, while still being edited by userA, userB sees the warning of edition conflict unless userB accepts to request to userA to enter into the collaborative session.
4. If userB accepts to request that coedition to userA, then UserA would see some warning about that request from userB in some popup/modal/or similar, with the option to accept and get his edition saved and restarted as online collaboration among both, or reject the request, and keep writing himself alone in that page.
5. therefore, save the edition in place by user A (the one that started editing the page first)
6. reopen that page for edition for user A with te url adapted to the togetherjs session, and
7. send user B to the edition session with togetherjs (or alternatively, show userB the same link to open the page for edition).
8. if userA accepted userB to write together, they both are working with TogetherJS on that page. And in that case, when other users such as userC clicks at edit the wiki pageFoo again, userA gets the same message requesting to confirm to accept new users getting the collaborative session, and when accepted, since the TogetherJS is already on, there wouldn't be any need to restart the edition session with togetherjs since it would be already in that mode.

This is currently partly working in 21.x LTS :

- user is sent automagically to open the session without "Warn on edit conflict"
- ... BUT the together session fails to display the other users being there editing the page concurrently:
 - therefore, a new version is saved, without noticing that the versions will not be seen by each other: `Edit Conflict triggered`.

To reproduce the updated problem info:

Come here with two different browser engines (or one in private browsing and hte other in normal, as usual to test this feature):

<http://xavi-9794-6160.show2.tikiwiki.org/tiki-index.php?page=HomePage>

user1:
u: admin
p: 12345

user2:
u: user
p: 54321

1. Click on button of the side module "Co-Write with TogetherJS"
2. Edit homepage as user1 (admin) in browser engine1 (e.g. firefox)
3. Edit homepage as user2 (user in browser engine2 (e.g. chromium)
 - accept the popup asking to join session

problem triggered: none can see that the other user is editing the page, saving, moving elsewhere, etc.



Solution

Fixed, while highlighter (codemirror) is disabled in the edit session (because codemirror is completely disabled site wide, or because the users willing to co-write with plugin together they all click at "disable highlighter in the wiki page edit session) .

Importance

9

Easy to solve?

6

Priority

54

Demonstrate Bug (Tiki 19+)

This bug has been demonstrated on show2.tiki.org

Ticket ID

6160

Created

Wednesday 02 November, 2016 18:34:16 GMT-0000
by Xavier de Pedro

LastModif

Monday 28 September, 2020 12:32:34 GMT-0000

Comments



Xavier de Pedro 01 Apr 20 10:10 GMT-0000

This is currently partly working in 21.x LTS:

- user is sent automatically to open the session without "Warn on edit conflict"
- ... BUT the together session fails to display the other users being there editing the page

concurrently:

- therefore, a new version is saved, without noticing that the versions will not be seen by each other: `Edit Conflict` triggered.

I'll add a show2.t.o instance asap so that it can be easily reproduced.

An urgent measure, of no potential quick fix is found, would be to disable the plugin together. Because it seems as if when plugin together is enabled (even if no module uses it), the warn on edit conflict doesn't take effect, and users are not warned that somebody else has that page open for edition already.



Victor Emanouilov 13 Apr 20 12:46 GMT-0000

Unfortunately, we depend here on `hub.togetherjs.com` as the standard `togetherjs` package is using mozilla-hosted hub server. This server seems to be down now (not sure how long but it seems at least a couple of weeks). See here <https://togetherjs.com/docs/#hosting-the-hub-server> for more info on how to host the hub server yourself. I see 3 possible ways to go ordered by easiest to hardest to implement.

1. Contact togetherjs team via irc or email and see what are their plans on supporting `hub.togetherjs.com`.
2. Host the server yourself somewhere where you host Tiki. It is a quite simple nodejs app that needs to be run and proxy'ed by an http server. If you choose this option, I can add a togetherjs option to provide the URL to the hosted hub server inside the wiki plugin.
3. Host the hub inside Tiki. Since we can't rely on a nodejs app in Tiki due to complex deployment issues we will hit, there is an option to rewrite the hub server in PHP and provide it as part of Tiki codebase. This will require some moderate programming effort to rewrite the nodejs app with proper PHP libs in plain PHP. This will be the cleanest solution for the long-term but require most effort now.



Xavier de Pedro 13 Apr 20 22:30 GMT-0000

Hi Victor, and thanks for your feedback. I wrote an email to the contact address shown in the `togetherjs.com` page

I had a look at 2), also, but was unclear to me in a quick look where that heroku app should be installed, and I didn't have the time to re-read the whole thing again (It would be my first thing with a heroku-thingy).

3) sounds interesting, in case it was reasonably sized task, even if I'd rather talk to Marc, then, to see his plans for Real Time collaboration in the mid and long runs. The nice thing about PluginTogether in Tiki is that it allowed to have already (since many tikis ago) some way to allow some real-time collaborative writing in text areas (like Linux kernel for the GNU OS vs. waiting for the GNU Hurd kernel?).

Thanks for sharing your thought on this topic, Victor, as usual.



Xavier de Pedro 14 Apr 20 09:21 GMT-0000

Oups, their contact email address `togetherjs@mozilla.com` is not valid anymore.

Regarding 2), I'd rather have a call (since I might grant access to a server where you could install

the hub server code - npm, etc.) to showcase the potential use of that solution. For instance, it could be at show2.tiki.org, which is already a server of mine devoted to Tiki uses. Would that fit in as a short term solution, while we seek a better/future-proof solution (option 3) if needed) with Marc (to find out whether the schedule of other RTC solutions would make 3) unnecessary)?

@marclaporte ? (I wonder if user mentions from comments work and sent a notification email to Marc)



luciaash d' being 14 Apr 20 13:10 GMT-0000

Xavier de Pedro Regarding the User Mentions and the notifications being sent: Yes it worked for me - although it seems to be dependent if the user puts their information public or private.



Marc Laporte 14 Jun 20 12:38 GMT-0000

Notifications: I usually watch all changes on a Tiki and filter messages to a folder. So maybe it worked but went to same folder.

About 1, 2, 3. (and 4)

- 1) We can't count on external services in general, and Mozilla stopped active support for this project. So it could end at any time.
- 2) This is least interesting unless there is a community improving. Should be available via Docker.
- 3) If we had done this a few years ago, we'd have a robust solution now. I think we should consider this seriously. Research PHP libs to see which ones we can leverage And we should make this available to others and get a community around this.

There is also #4 which should considered first. Use XMPP (Openfire) and WebRTC (Jitsi Meet). TogetherJS has overlap like the audio.

I remember Dele ported the server-side code to XMPP for Openfire.

If #4 not suitable, fallback on#3.



Marc Laporte 14 Jun 20 13:09 GMT-0000

But

<https://github.com/jsfiddle/togetherjs/issues/1172>



Xavier de Pedro>Marc Laporte 27 Jul 20 14:18 GMT-0000

Replied to But...

Thanks Marc. Today Victor added a new param serverurl to PluginTogether in 21.x and

master/trunk,
http://doc.tiki.org/PluginTogether#Param_serverurl

Victor confirmed that the server url from that post you linked works:
<https://togetherjs-hub.glitch.me/>

In addition, we tested how to setup the server code and we documented the steps on how to install the server side code in an ubuntu-based server (where r.tiki.org is hosted).
http://doc.tiki.org/PluginTogether#Setup_up_your_own_server

Let's re-test that we can use again PluginTogether in production (because we need that RTC feature up & running)

HTH



Jonny Bradley > Xavier de Pedro 27 Jul 20 14:42 GMT-0000

Replied to Thanks Marc.... Well done and thanks all, looking forward to trying it again!

Attachments

filename	created	hits	comment	version	filetype
----------	---------	------	---------	---------	----------

No attachments for this item

The original document is available at
<https://dev.tiki.org/item6160-PluginTogether-should-replace-the-url-associated-with-the-Edit-page-button-s-o-that-co-editors-are-automagically-offered-to-go-to-the-session-with-collaborative-edition-instead-of-the-warning-of>