See also Realtime

Plugin Together

This wiki plugin is an experimental service for your website from Mozilla Labs called TogetherJS, that makes it surprisingly easy to collaborate in real-time: notify unique urls, co-write, talk, follow pages visited by your buddies. It was formerly known as TowTruck. It was added in Tiki12 and improved in Tiki15, Tiki19 and Tiki21.

Since Tiki22 you can specify a different server through a new parameter added to the plugin, in case the server side code needed for this plugin to work is no longer working with the default server provided by Mozilla.

You can easily use this plugin in your Tiki installation by means of applying the Together profile.

**Note:** Together is alpha-quality software. We do not recommend using it in production at this time, even if it looks promising as a Real Time Collaboration (RTC) tool.

### Parameters

*Introduced in Tiki 12.*

*Go to the source code*

*Preferences required:* wikiplugin_together

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Accepted Values</th>
<th>Description</th>
<th>Default</th>
<th>Since</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>buttonname</code></td>
<td>text</td>
<td>Set the button name. Default is CoWrite with TogetherJS</td>
<td>CoWrite with TogetherJS</td>
<td>12.0</td>
</tr>
<tr>
<td><code>serverurl</code></td>
<td>text</td>
<td>Hub server URL address if the default one is not working or you are willing to host your own hub server.</td>
<td></td>
<td>21.0</td>
</tr>
</tbody>
</table>
A service for collaborating on your website in real-time. TogetherJS lets users communicate, co-author, co-browse and guide each other. TogetherJS is implemented in JavaScript; no software or plugins to install, and it is friendly with existing web pages, while still letting developers customize the experience. Note: TogetherJS is alpha-quality software. We do not recommend using it in production at this time.

Current Limitations

- Before Tiki15: You need to deactivate "Warn on edit conflict" in Settings -> Wiki -> General preference Otherwise, you will get this annoying message: "Override lock and carry on with edit"
- If you are permitting anonymous edits, you’ll want to turn off "Anonymous editors must enter anti-bot code (CAPTCHA) " Settings -> Security -> Spam protection

Where does it work?

In theory, TogetherJS can work with any feature in Tiki. However, some features in Tiki may need adjusting. The following table is to keep track of the state of each feature.

<table>
<thead>
<tr>
<th>Version</th>
<th>Feature</th>
<th>Works or not</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2</td>
<td>Wiki pages with wiki syntax</td>
<td>yes</td>
</tr>
<tr>
<td>12.2</td>
<td>Spreadsheet</td>
<td>no</td>
</tr>
<tr>
<td>12.2</td>
<td>Tracker Items</td>
<td>yes (textarea)</td>
</tr>
<tr>
<td>12.2</td>
<td>Drawing with SVG-edit</td>
<td>No</td>
</tr>
<tr>
<td>12.2</td>
<td>Slideshow in view mode</td>
<td>No</td>
</tr>
<tr>
<td>12.2</td>
<td>Slideshow in edit mode</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Examples

This code added to a custom module, or a module menupage in a side column:

```
{together}
```

Would produce a TogetherJS button. Once clicked by a user, it will add a TogetherJS widget in the right margin of the window, as in the following image:
In this example,

1. user Mary has clicked at the button "CoWrite with Together" (formerly known as "TowTruck", like in the screenshots)
2. the TogetherJS widget is shown up in the right margin of the window
3. Once you click at the icon to invite friends, you will get a unique url that you can send over email, smm or any means to your buddies, so that they can start cowriting with you in that Tiki page (tracker item in this example)
4. You can also change your default username in Together, set your own avatar, color, etc. In this example Mary sets the avatar to a puppy, and keeps the default brownish-redish font color.

- Additionally you can talk (experimental) or chat with your buddies.

Once you sent the link to your buddy (let's say, user John), and he clicks on it, he will see a window like the following requesting him to accept the collaborative session:

Unfortunately, the message in the window is not visible in the screenshot, but it could look something like this:

```
Join TowTruck session?

Your friend has asked you to join their TowTruck browser session to collaborate in real-time!

Would you like to join their session?

No, don't join
Yes, join session
```

Once accepted, he can log in that Tiki site (if needed), and he will see something similar to Mary.
Clicking at the avatar of your user you will be able to see the options to rename your user, set avatar, font color, etc.

In this example, John renamed his Together username accordingly, choose a tennis ball avatar and did set his font color to orange. After that, he continued editing the contents of the tracker item, that became updated in real time in Mary's browser also.
At some point during 2019 PluginTogether stopped working because some server side software was missing from the default server from mozilla labs. During July 2021 a new param was added to this plugin in Tiki21 in order to allow using other server urls (param `serverurl`), such as

https://togetherjs-hub.glitch.me/

Alternatively, you can set up your own server for togetherjs server-side code.

Setup up your own server

See: https://togetherjs.com/docs/#hosting-the-hub-server

On Ubuntu 16.04 servers, you need to:

1. install npm (npm v 3.5.2; node version 4.2.6 are the default versions, and they seem to be ok for PluginTogether to work)
   
   ```
   sudo apt-get install npm
   ```

   You also need to:

2. clone togetherjs code
   
   ```
   sudo mkdir -p /var/www/togetherjs/ # assuming /var/www is not the public apache webroot but /var/www/html/
   cd /var/www/togetherjs/
   sudo git clone https://github.com/jsfiddle/togetherjs.git
   ```

3. install node dependencies
   
   ```
   cd /var/www/togetherjs/
   sudo npm install
   ```

4. make a symbolic link from /usr/bin/nodejs to /usr/bin/node
   
   ```
   sudo ln -s /usr/bin/nodejs /usr/bin/node
   ```

4. run server binding it to a domain, eventually with https (ideally), using apache proxy to server the node server in port 8080 to users visiting that domain with http or https.
   
   1. enable mod proxy proxy_http proxy_wstunnel
      
      ```
      sudo a2enmod proxy
      sudo a2enmod proxy_http
      sudo a2enmod proxy_wstunnel
      ```
5. make the apache virtual configuration file

```bash
cd /etc/apache2/sites-available/
sudo cp example0.org.conf example1.org.conf
sudo nano example1.org.conf
sudo a2ensite example1.org
sudo service apache2 restart
```

- Example contents of this `/etc/apache2/sites-available/example1.org.conf`

```html
<VirtualHost *:80>
  # The ServerName directive sets the request scheme, hostname and port that
  # the server uses to identify itself. This is used when creating
  # redirection URLs. In the context of virtual hosts, the ServerName
  # specifies what hostname must appear in the request's Host: header to
  # match this virtual host. For the default virtual host (this file) this
  # value is not decisive as it is used as a last resort host regardless.
  # However, you must set it for any further virtual host explicitly.
  #ServerName www.example.com
   ServerName example1.org

   ServerAdmin webmaster@localhost
   DocumentRoot /var/www/html

   RewriteEngine on
   RewriteCond %{HTTP:Upgrade} websocket [NC]
   RewriteCond %{HTTP:Connection} upgrade [NC]
   RewriteRule .* "wss:/localhost:8080/$1" [P,L]
   ProxyPass / https://localhost:8080/
   ProxyPassReverse / https://localhost:8080/
   ProxyRequests off

<Directory /var/www/html/>
   Options Indexes FollowSymLinks MultiViews
   AllowOverride All
   Require all granted
</Directory>

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

   ErrorLog ${APACHE_LOG_DIR}/error.log
   CustomLog ${APACHE_LOG_DIR}/access.log combined

   # For most configuration files from conf-available/, which are
   # enabled or disabled at a global level, it is possible to
   # include a line for only one particular virtual host. For example the
   # following line enables the CGI configuration for this host only
   # after it has been globally disabled with "a2disconf".
   #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

`# vim: syntax=apache ts=4 sw=4 sts=4 sr noet`

---

Run the server with

```
cd /var/www/togetherjs/; node hub/server.js &
```
# or sudo node hub/server.js --host=server_ip
# replace server_ip with the ip of this server where you installed this software
(example1.org in this example)

Check the process with:

```
ps aux | grep node
ps aux | grep hub
```

- http://example1.org/status > should say "OK"

Related links

- https://togetherjs.com/
- https://hacks.mozilla.org/2013/10/introducing-togetherjs/
- http://www.ianbicking.org/blog/2014/03/towards-next-level-of-collaboration.html
- https://www.ianbicking.org/blog/2014/09/professional-transitions.html

Aliases

PluginTowTruck | Together | Mozilla Together | Mozilla TowTruck | TogetherJS