How to use Twist Collaboration tool.

Oct 17, 2018

It became clear to me that the tools isn’t as intuitive as I thought, so this is a short outline of how I think we could use it, and what might make it better.

1. On the left of the screen you'll see "channels" I like to think of those as "rooms" - These are what I think should be used for specific collaboration on projects. IE: "Maqueen Robots" channel would be for collaboration on that project.
2. Inside each "channel" is threads, the #General channel is just everything in general.
3. If you click on a channel - in the middle of the screen, you should see topics inside that channel - called threads, near the top you'll see start a new thread. Threads are messages that have a theme in common.
4. Members will have the ability to create new channels, and new threads inside of channels. They can invite other members, join and browse all of the channels (that are marked public).
5. A Guest can only access the channel they have been invited to by a member or admin.
6. Messages - These are (as far as I can tell) updates within some channels, and private messages between members, and team mates.
7. The #General Channel could be/should be used to just share some updates, and other channels can be created for larger projects, each with its own threads - and members to that channel (teammates).
8. If you want to email a “Channel” you need the email link, this can be found by clicking on the “channel” you want to email - in the center of the screen you’ll get the channel name, with a icon that has three circles (these are the settings for the channel), one of the selections from the menu is “Get email to channel….” click this and you’ll be given a link.
   1. I believe the link will be different for each member, this way twist knows who sent the message.
   2. Email appears to be incoming only - it doesn’t appear to send emails back to the address that was used to sent the original message.
   3. A email message can come from a email address that wasn’t used to create or setup your twist account (This is why I believe each email link is unique per member)
   4. A email that is sent creates a new Thread.
9. Notifications - I’m still trying to figure these out, I get some and others I don’t get.
   1. If you click on your picture or icon, in the upper right hand corner of the screen you’ll see settings, one of the tabs is “My Notifications”. I’ve got mine set to notify on both messages and threads.

I have setup a couple of test channels one is for testing Github integration, the other is for Todoist integration.

1. Github integration works really well, but isn’t as straightforward to setup as it should be, it’s not hard, but the instructions miss a step, so ask, and I can help :-)
2. Todoist - I still have a lot of questions about how this works, and if a todoists is shared to everyone, or if it just goes into my personal todoist lists. (lots of questions here).

# Github Pages

As mentioned, a few times, Github provides some decent tools for project documentation.

Github Pages is one of those tools, it is super easy to setup, just follow the guide here.

<https://blog.github.com/2016-12-09-publishing-with-github-pages-now-as-easy-as-1-2-3/>

This is by far the easiest way to create a project document, a website is generated that can be shared, or linked.

IF a markdown file is edited, the new site also updates.

There are some tools available to make some nice websites.

1. <https://prose.io> and online editor for basic markdown, that updates and integrates with github, submitting the changes as commits, and pushing the updates when saved.
2. MkDocs - <https://www.mkdocs.org/> fast, simple, and downright gorgeous static site generator that geared towards building project documentation. Written with markdown, and configured with a single YAML configuration file. This is software that you download and use from the CLI of your machine, the generated site can be hosted anywhere. It also integrates with github, and github pages nicely. It is a little more complex to use than just a the simple 123 github pages method above, but the sites it generates can be a lot more complex. The User Guide section of the their site has everything you need to know to make sites (and it’s pretty simple to use).

# GITHUB Learn by Doing repository.

Created maybe a year ago, this was meant to be used to learn a bit about how to use github.

<https://github.com/automation-technology-club/Github_Learn_by_Doing>

Today, I created a site using the really simple (easy as 1234) method above, after a site was created, I used <https://prose.io> to edit and update it. I even gave it a nice theme, and it looks amazing for something that took all of 2 minutes to create.

<https://automation-technology-club.github.io/Github_Learn_by_Doing/>

A couple of things here - you will notice the URL is different. This site was generated using just the README.md file that was already in the repository.

Last thing - You can see a complete commit history for this repository, including the changes made to the README.md file here: <https://github.com/automation-technology-club/Github_Learn_by_Doing/commits/master>

Why is this important, because we have a complete record of everything we have done, changes, and (hopefully) why we made those changes. IF we break something, we can go back in time to a point that it isn’t broken, and try to figure out what we did to break it. In the case of documentation, it mean that if we add a feature, and later remove that feature, we have a record of what that feature did (if we go back in time).