Tiki and Git, fork and merge request

Tiki and Git, fork and merge request

 Bernard Sfez - 2020-06-12 21:10



In this Tiki Express tutorial we'll review together;
*How to create your project at Gitlab
*A quick explanation about Security and Gitlab
*How to mirror the Tiki repo in your project
*An overview of the workflow
*How to install a local Tiki
*How to create your branch
*How to commit your code
*How to push it into your fork
*How to select you default branch
*How to create a merge request on the Tiki repository

Hello Internet, I'm Bernard Sfez a Tiki Specialist and in this tutorial we'll see together how to work with Tiki and Git the easy way using your fork and creating a merge request.
This tutorial is aimed for developers new or skilled, designers, translators or anyone willing to contribute to the Tiki project.
Hang on, it will be a speedy tutorial.

And first of all I hope that you are all safe and things are okay for you and your family.

Secondly I have to apologise for not being able to publish earlier this Tiki Express Tutorial.
As you've seen in the pre-intro I tried several time to start it but got so many thing in the way I just couldn't drop anything else to complete it and that's the way it is.

I also had to switch the topic of the video as announced in the previous tutorial for a shortened one but no worries there will be soon one about the custom search plug-in in Tiki.

Now some quick news.

Tiki 21 has been released and it is an awesome release. Check details at https://doc.tiki.org/tiki21. And you guess right the link will be in the description below.

We have a Tikifest coming and this year it will be a virtual meeting. I don't need to tell you that anything involving traveling is very complicated but Tiki is a perfect tool, a perfect application for online collaboration and working remotely.
Therefore the Tiki community organised and invited to participate and contribute to our 2020 virtual Tikifest. Check details again link in description, I won't say it again.
https://tiki.org/TikiFestVirtual2020Last but not least following requests and wishes and discussion on how I can optimise the help
I can provide about Tiki I decided to held a weekly open discussion.
it's an unscripted information sharing opportunities and if you want to receive my meet-up notifications you will have to go to my website and subscribe to the newsletter, no other way.

And now we start.

Tiki is an open source application and exists since 2004 with an original model that kept us on top of the CMS, Wiki Web, collaborative platform and Web application generator.
Tiki is free and everyone can join the community and start to contribute right away as long as you follow our three rules (https://dev.tiki.org/3-Rules):
*Respect the environment,
*Commit early commit often,
*Make it optional,

To support Tiki we are using Git and a Tiki repository at Gitlab.
The Tiki community decided to use Gitlab to develop the Tiki project and it means it also store and distribute the files from the project in a repository. To be able to follow this work properly, you need to have a Gitlab account or to create one at Gitlab.com.
(https://dev.tiki.org/Git-workflow-For-Collaborators)

A quick explanation about security and Gitlab

Let's open a discussion about security at Gitlab.

One important thing with your account and project is to keep it safe and most of the time private. This is done through the security settings of your accounts. Gitlab has serious and justify concern about security and password. I suggest you to register your SSH key following the Gitlab doc and to activate double factor authentification. I strongly suggest you to take the time to read their manifest about security practices.
(https://about.Gitlab.com/handbook/security/)

How to mirror the Tiki repo in your project

Once you have your account create a project so you can create your Tiki fork and mirror it with the original Tiki repository. That way your project will be constantly updated using the Tiki repository and you will be always working on updated files. On Gitlab go to setting repository and expand the "mirroring repositories". From here copy/paste the git URL from the Tiki project.
(https://Gitlab.com/tikiwiki/tiki.git)

Enable "overwrite divergent branches" and save.

An overview of the workflow

You may have already some concern.
What if you modify files, won't there be over written?
With the new code coming in from Tiki ? Would they be merged ?
Are my modifications pushed back to the Tiki repository ? Etc.

Well while there are more than one reason to work with Git I'll explain here our community model and in fact the reason you are working on your fork and create your branch to submit your merge request.

In short:

- You create a branch in your fork to modify and checkout your code
- You push it back to your branch in your fork your repository
- You submit a merge request to the Tiki project so others can review your code, ask questions… #Merge your requestsAt Gitlab I go on my project page. As you can see by default the first version of Tiki is selected and it will be used for comparison. To avoid confusion I suggest you to set your default work branch with the one you are using. As habit and safe practice I always select the default branch I'm sending a merge request to.
  Go to "Settings", "Repository" and open the "Default branch" section.
  Select the branch you want and save.
  It is done.

I have a repo, I have my fork of Tiki and I will clone my first branch from my remote fork on my local computer. By the way I've already made a much more dense video about Git and Tiki where you can find more details about striping history, selecting your branch etc.You can check it or search the web to investigate further.

How to install your local Tiki

So I will create a lightweight clone of the branch 21.x with the minimum history from my repo on my local. On my local using my fork it will look like this:

```
git clone --depth=1 --branch=21.x https://Gitlab.com/mycloneurl.git
```

I go on my terminal and I enter it.Now in real life I should complete the Tiki setup, test everything that I do and be sure my local is working fine before committing anything. But this is a demo and I'll skip these steps but you, you don't do that at home. Before committing check that everything is working fine and does not break anything else in Tiki.

How to create your branch

We will now create a new temporary branch that will be used to commit a little and simple

addition to the Tiki French translation language.

On my local terminal I check the existing branch I have.

```
git branch -a
```

I create usually a branch per fix so I can submit and control my commits and merge request. Let's create a branch "french-translation".

```
git checkout -b french-translation
```

I check again my local branches and I can see my new branch has been created and it is the one I'm working on.

How to commit your code

Another good practices is to rebase, check that you use the last version of the code before starting to work so you don't have during the commit operation conflicts or merge to manage.

```
git pull —rebase
```

Now I quickly edit the translation file I pick up a string to translate and save.
I'm ready to commit but before that I remind you it is important to double-check your Tiki first.
Not less important you should regularly check the "How and where to commit information" at dev.tiki.org.
To commit my file the first time I need to add this file to prepare the content staged for my next commit.

```
git add lang/fr/language.php
```

Then I create my commit and add the commit message the way it should look.

```
git commit -m "[TRA] Multilingual: French translation"
```

I added here the proper tag, the feature it concern and the description of the commit

How to push it into your fork

Creating a commit is a local operation and as long as you don't push it it won't affect the remote repo or the Tiki repository. It is possible to check the pending commits using this command and what we see here is what was expected.

```
```

```
git log --branches --not —remotes
```

To push my modified branch into my fork I do.

```
git push -u origin french-translation
```

I got a nice message that show that it has been done. Now I want to have my commit merged into the Tiki repository.At Gitlab I go on my project page, now I go on the Gitlab menu and I click on "Repositories", "Branches". There I can see the new branch I created with the commit number and the status of the changes between version. This comparison is done using the default branch that we set.
The numbers should be very low here if you have higher numbers it may be the sign that you are not doing the right thing and you should double check.

At Tiki we use a CI/CD pipeline to automate code review and test. You may have to wait a little time for these to complete before being able to complete your merge request. If we dive into our commits we see some warnings but while you can check what they are about warnings are just wanting and depending on what you are committing and expect, it may be perfectly normal.
In our case we are just good to go.

How to create a merge request on the Tiki repository

So I finally click on the merge request button check from branch to branch information and eventually change them by clicking on the "Change branches" link.

I pasted or right down the same message I used for my commit message I give a description with detail of what behavior was changed or fixed, how to reproduce etc. Anything that can help others to review quickly the merge request. Here a sample of what I usually do to illustrate as for this merge request, a translation, there is not really much to say and to explain.

As you can see I also recreated on my project the same tags that we are using in the Tiki repo and so it is easier and more user friendly to identify the nature of my merge request.

I have also some options so I can delete the source branch when the merge request is accepted.
I "Submit the merge request" and now I'm redirected inside the Tiki repository where I can see my merge request for my multilingual French translation, I click on the "Merge" button in that case because there is anything else I would like to comment (about this commit).

Now as you can see we have a branch and it is updated with the last commit.Also note that you have on the left side a merge request menu that you can always use to comment and discuss or see pending more complex commits that would require approval or if you want to share and have some guidance when you commit something.

Goodbye

No more excuses !
You can contribute without fear to break anything and you can have your code available for

others to review and eventually help.

I hope you've enjoyed this tutorial as usual. Let me know in
the comments if you liked it. Like and Share this video of course.I won't forget to publish one about the customSearch plug-in and there will be more video tutorials on the plugin list listExecute and the trackers. If you don't want to miss my next Tiki Express Tutorial, subscribe and ding ding on the little YouTube.Don’t forget I answer, I will answer questions on my open chat weekly meeting so subscribe at bsfez.com to the newsletter and you will be you will receive my meet-up notifications.

That's it for today and may the power of Tiki be with you...

Bloopers

Well... I’d said it was quite right. ?

Interesting links

https://doc.tiki.org/tiki21
https://dev.tiki.org/tiki21
https://dev.tiki.org/3-Ruleshttps://dev.tiki.org/Git-workflow-For-Collaborators
https://Gitlab.com
https://about.Gitlab.com/handbook/security/
https://docs.Gitlab.com/ee/ssh/
https://Gitlab.com/tikiwiki/tiki.git
https://about.Gitlab.com/handbook/security/
http://dev.tiki.org/Where-to-commit
http://dev.tiki.org/Commit-Tags
https://youtu.be/2vCBhfzSe7Q

Command used

git clone --depth=1 --branch=21.x https://Gitlab.com/bsfez/yourprojecturl.git .
git branch -a
git checkout -b french-translation
git pull —rebase
git add lang/he/language.php
git commit -m "TRA Multilingual: French translation"
git push -u origin french-translation
git log --branches --not —remotes