

# Tech Reference

- [Git Cheat Sheet / Guide](#)
- [Setting up Obsidian Sync for GameDev](#)

# Git Cheat Sheet / Guide

<https://ittools.fifthdread.com/git-memo>

## Install Git on MacOS

```
brew install git
```

## Install Git on Windows

<https://git-scm.com/download/win>

## Set Config

```
git config --global user.name "[name]"  
git config --global user.email "[email]"
```

example:

```
git config --global user.name "testuser"  
git config --global user.email "email@emailtest.com"
```

## Set Editor to nano on MacOS

```
git config --global core.editor "nano"
```

```
export GIT_EDITOR=nano
```

```
export VISUAL=nano  
export EDITOR="$VISUAL"
```

## Generate SSH Key

## On Windows

```
ssh-keygen.exe -t rsa -b 4096
```

## On MacOS / Linux

```
ssh-keygen -t rsa -b 4096
```

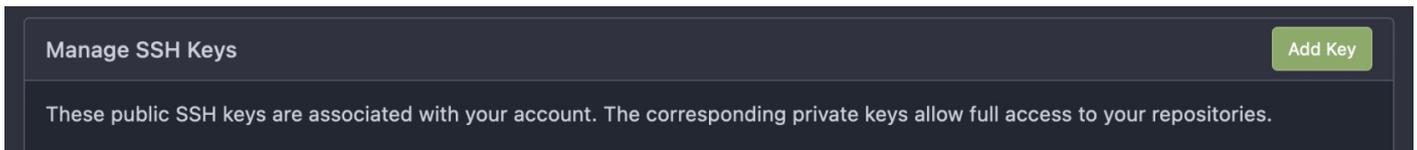
Press Enter - No Passphrase, leave empty

# Take public key and add into Gitea

Key is located in `/Users/<username>/.ssh`

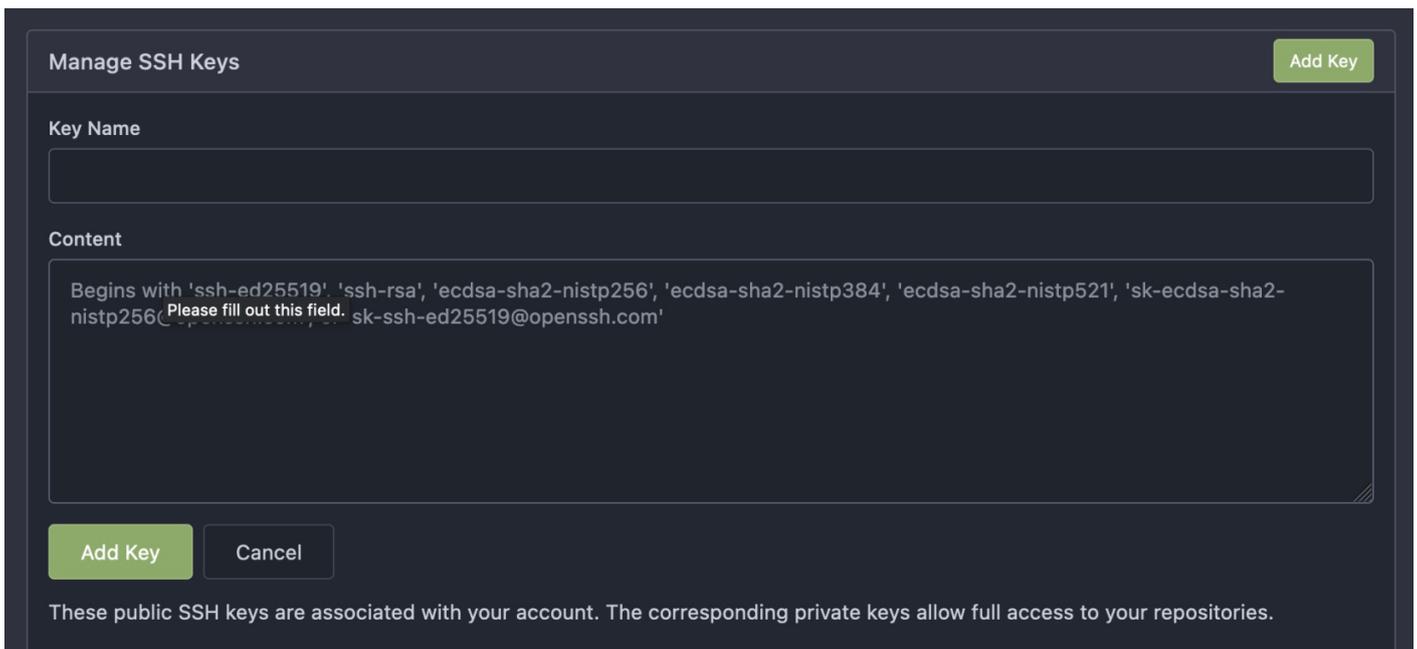
Take `id_rsa.pub` - Open in text editor. Copy contents.

Go to profile on Gitea HERE: <https://gitea.fifthdread.com/user/settings/keys>



## Add Key

Paste public key information into Content Box.



It should auto-complete the Key Name. Click Add Key to complete.

Your machine can now be trusted to access the Git Repository via SSH. You can now pull and push to the repo.

## Cloning a project / Pulling a Project

To Clone a project from the repo, go to the project on gitea.

Switch to SSH and click the Copy button to copy the SSH access URL.

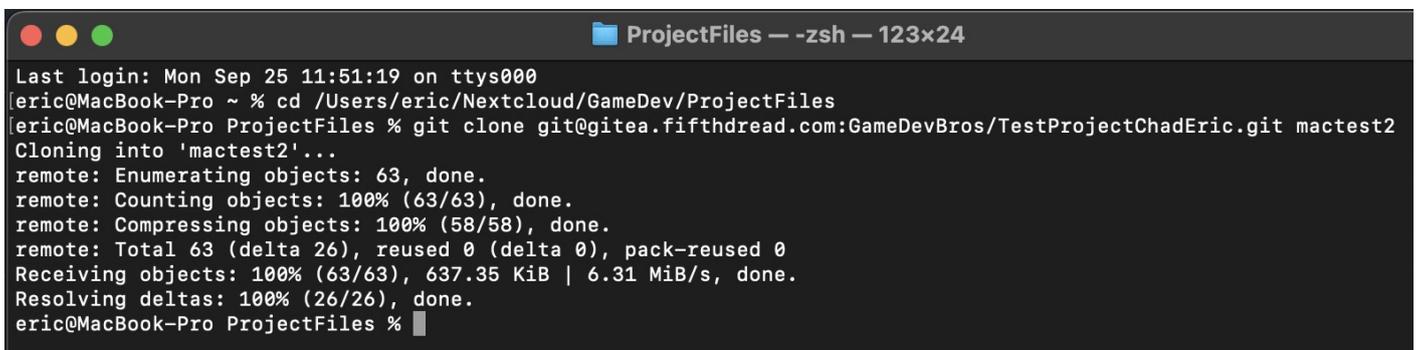


Navigate to the directory you want to clone your project into. In the example below, I go to a ProjectFiles folder.

Type `git clone "URL" "DirectoryName"`

```
git clone <PASTE URL HERE> <BRANCH NAME HERE>
```

The branch name is not required. It is simply the name of the directory you are going to use for the files. I like to stay organized and make it the name of the branch.

A terminal window titled 'ProjectFiles — -zsh — 123x24'. The terminal shows the following output:

```
Last login: Mon Sep 25 11:51:19 on ttys000
eric@MacBook-Pro ~ % cd /Users/eric/Nextcloud/GameDev/ProjectFiles
eric@MacBook-Pro ProjectFiles % git clone git@gitea.fifthdread.com:GameDevBros/TestProjectChadEric.git mactest2
Cloning into 'mactest2'...
remote: Enumerating objects: 63, done.
remote: Counting objects: 100% (63/63), done.
remote: Compressing objects: 100% (58/58), done.
remote: Total 63 (delta 26), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (63/63), 637.35 KiB | 6.31 MiB/s, done.
Resolving deltas: 100% (26/26), done.
eric@MacBook-Pro ProjectFiles %
```

it will now download the repo.

## Renaming your local git branch

This is important, as if you do not rename your branch prior to pushing it to the repo, it will override the main branch. Not good!

Check what branch you are on, and what the branch name is:

```
git branch
```

Rename Main branch to something else via the below command:

```
git branch -m <NAME>
```

```
[eric@MacBook-Pro another_mac_test % git branch
* main
[eric@MacBook-Pro another_mac_test % git branch -m another_mac_test
[eric@MacBook-Pro another_mac_test % git branch
* another_mac_test
eric@MacBook-Pro another_mac_test % █
```

## Branches

To upload changes to a project, make sure you are working on a separate branch- RENAME YOUR BRANCH

Never upload directly to the main branch on a collaborative project, as branch merges need to be reviewed by the team prior to being merged.

## Committing Changes

First verify you have renamed the Branch

```
git branch
```

If it is something other than main, we can proceed.

```
[eric@MacBook-Pro mactest5 % git branch
* anothermactest
```

To see the changes you made since the repo was pulled, type:

```
git diff
```

```
[eric@MacBook-Pro mactest5 % git diff
diff --git a/Buttontest.gd b/Buttontest.gd
index 42d6c86..6532978 100644
--- a/Buttontest.gd
+++ b/Buttontest.gd
@@ -7,6 +7,7 @@ extends Button
 # comment amanda test
 # comment on macbook test
 # comment test on mac again
+# comment another test! On mac.

var image: Sprite2D

eric@MacBook-Pro mactest5 %
```

Commit the changes with the following command:

```
git commit -a
```

```
eric@MacBook-Pro mactest5 % git commit -a
```

The NANO editor will appear if on MacOS, or your preferred editor on Windows, such as Notepad++. Edit the top of the document with your commit comments. Save when complete.

If you opened in VIM and not nano, it looks like this... ew. Don't ask me how VIM works.

```
mactest5 — vi ◀ git commit -a — 80x16
This is a commit of my recent changes.
#
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch another_mactest
# Your branch is up to date with 'origin/main'.
#
# Changes to be committed:
#   modified:   Buttontest.gd
#
~
~
~
~
-- INSERT --
```

This is what it should look like if NANO opens the file.

```
mactest5 — nano ◀ git commit -a — 80x24
../d/GameDev/ProjectFiles/another_mac_test/mactest5/.git/COMMIT_EDITMSG Modified
This is a commit comment test!
#
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch another_mactest
# Your branch is up to date with 'origin/main'.
#
# Changes to be committed:
#   modified:   Buttontest.gd
#
^G Get Help   ^O WriteOut  ^R Read File ^Y Prev Pg   ^K Cut Text  ^C Cur Pos
^X Exit       ^J Justify   ^W Where is  ^V Next Pg   ^U UnCut Text ^T To Spell
```

CTRL+X to exit, and yes to save

```
[eric@MacBook-Pro mactest5 % git commit -a
[anothermactest 4eee5cb] This is a commit comment test!
1 file changed, 1 insertion(+)
eric@MacBook-Pro mactest5 %
```

This indicates success.

To check the status of your branch.

```
git status
```

```
[eric@MacBook-Pro mactest5 % git commit -a
[anothermactest 4eee5cb] This is a commit comment test!
1 file changed, 1 insertion(+)
eric@MacBook-Pro mactest5 %
```

## Uploading your new Branch / Pushing Changes

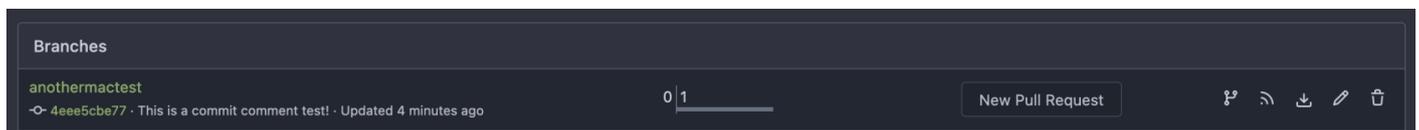
Again, make sure you aren't on main branch.

```
git branch
```

Now type in the following to upload your changes!

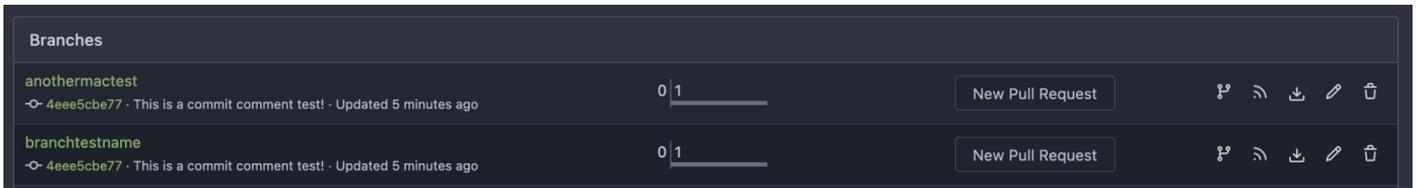
```
git push origin HEAD
```

This will upload a branch based off the name defined in "git branch"



Alternatively, define a branch name in the command.

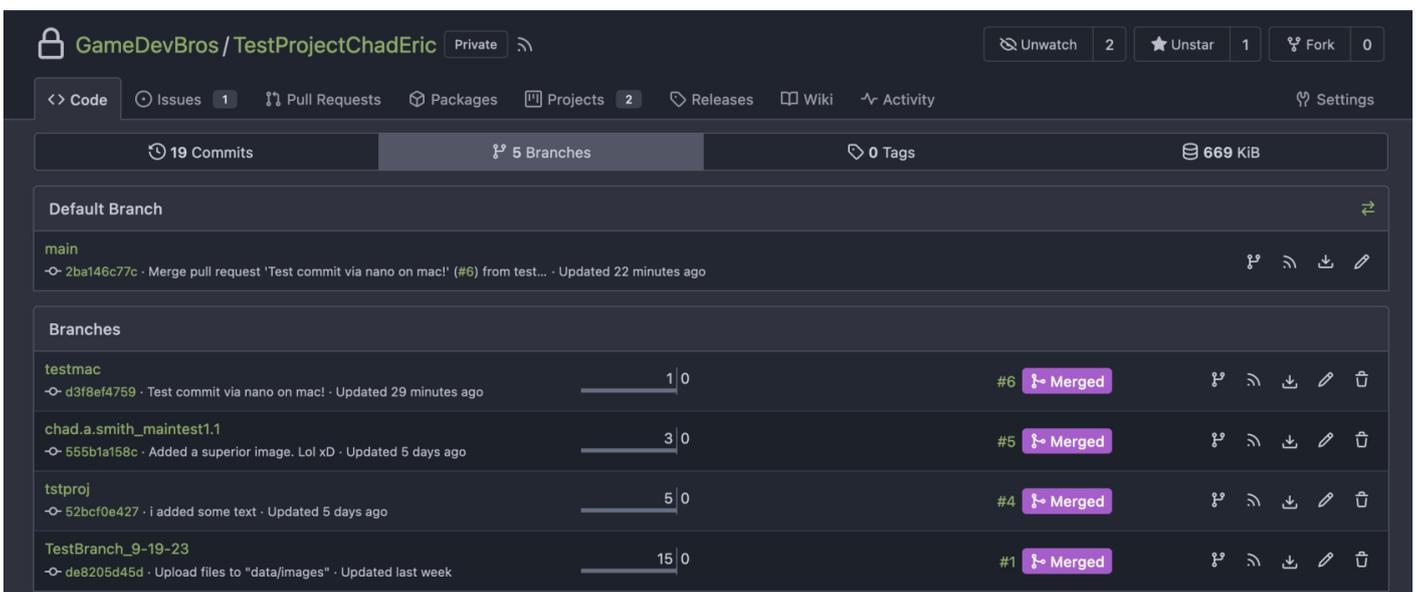
```
git push origin HEAD:branchtestname
```



Done!

## Looking at different branches on Gitea

Go to Branches. Click on New Branch from Main.



In this example, I made a test branch called another\_mac\_test

# TestProjectChadEric

Manage Topics

 19 Commits

 another\_mac\_test ▾



Go to file

Add File ▾



Fifthread

2ba146c77c

Merge pull request 'Test com



data/images



.gitattributes

See how it is selected here- it shows "another\_mac\_test" indicating we are viewing the new branch.

## Submit Pull Request - Merge with Main

When you have made changes and want to pull them into the main branch. Open Gitte, go to branches, and click New Pull Request on your branch.



You will open a pull request with all your changes outlined. Click on New Pull Request again.

GameDevBros / TestProjectChadEric Private

Unwatch 2 Unstar 1 Fork 0

Code Issues 1 Pull Requests Packages Projects 2 Releases Wiki Activity Settings

### New Pull Request

Select the branch to merge into and the branch to pull from.

merge into: GameDevBros:main ... pull from: GameDevBros:anothermactest

New Pull Request

1 Commits main ... anothermac

Author	SHA1	Message	Date
Fifthdread	4eee5cbe77	This is a commit comment test!	9 minutes ago

1 changed files with 1 additions and 0 deletions

Buttontest.gd

```
1 @ -7,6 +7,7 @@ extends Button
2
3 # comment amanda test
4
5 # comment on macbook test
6
7 # comment test on mac again
8
9 + # comment another test! On mac.
10
11
12 var image: Sprite2D
13
```

Unescape View File

You will now comment on the pull request, then click Create Pull Request.

This is a commit comment test!

Start the title with **WIP:** to prevent the pull request from being merged accidentally.

Write Preview

H B I Aa ↺

This is a detailed comment on why it should be pulled into main!

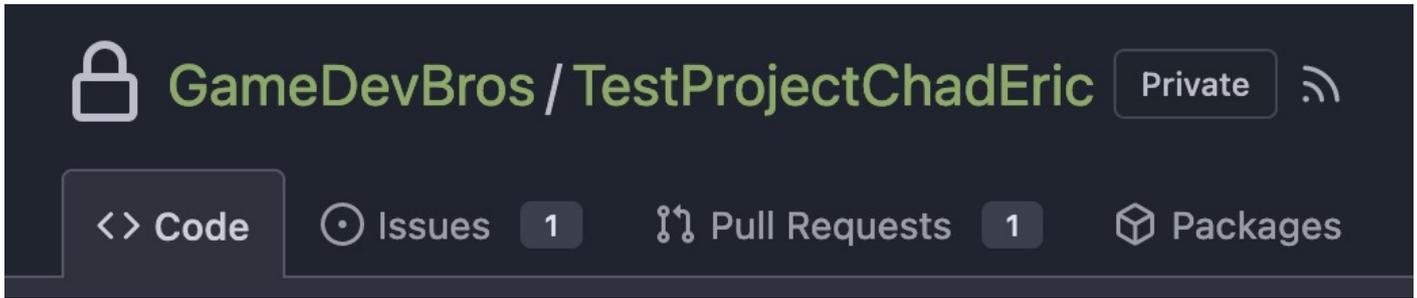
Drop files or click here to upload.

Create Pull Request

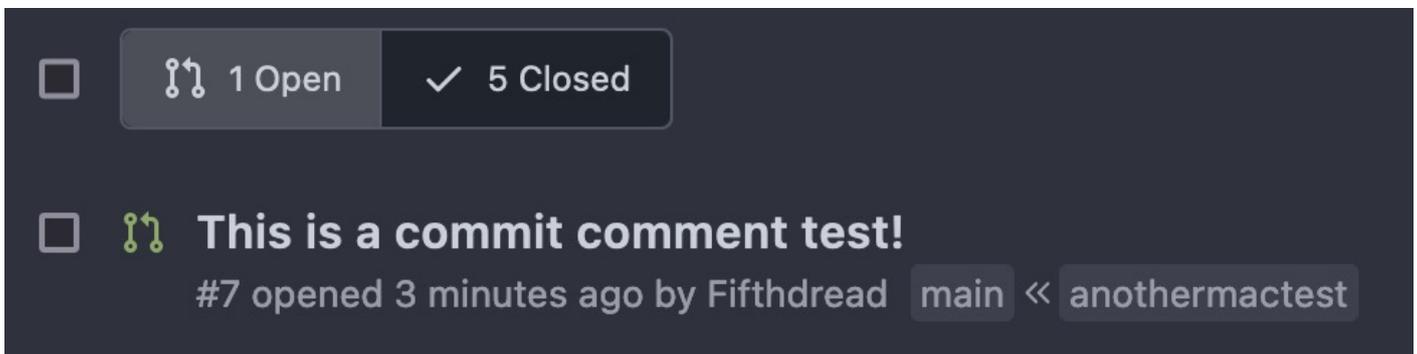
The request will now be reviewed.

# Approve Pull Requests

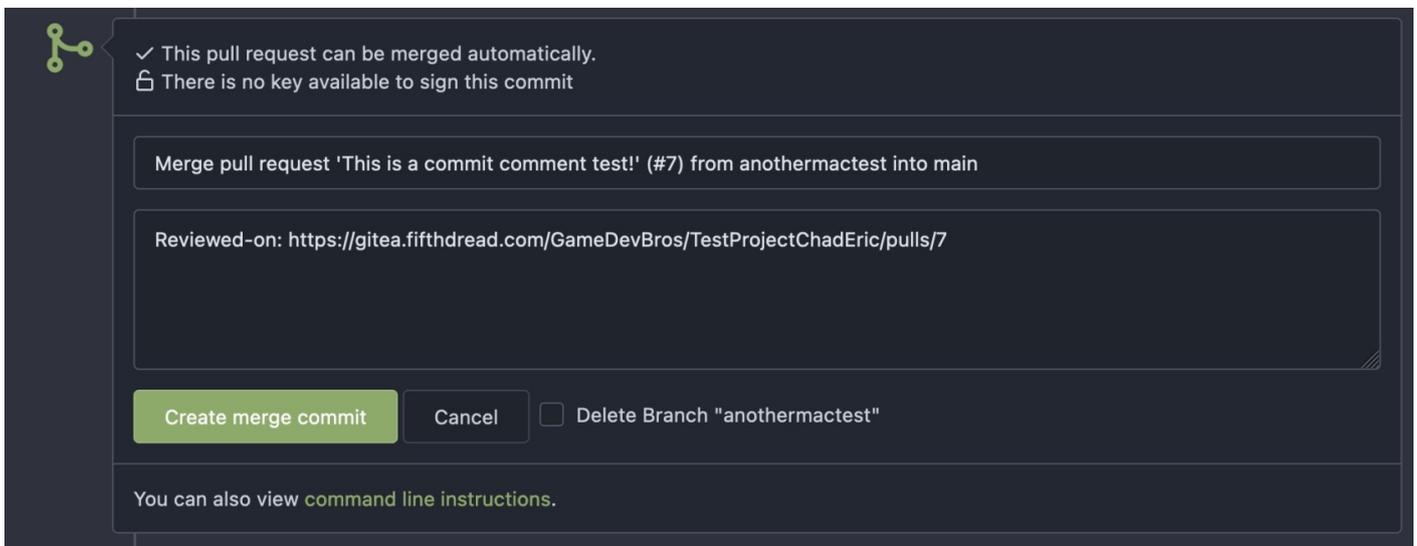
Pull Requests must be reviewed prior to merging. You can see the "1" indicating 1 request is waiting for review.



Clicking on Pull Requests shows the waiting requests.



This pull request can now be reviewed and merged.



The branch can also be deleted after it is merged with the main branch successfully.

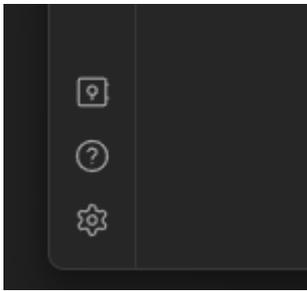
You're done!

# Setting up Obsidian Sync for GameDev

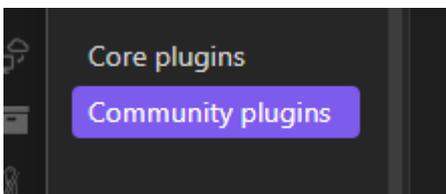
[Download Obsidian on any device](#)

Create a Vault on your device. Name it anything. I name mine GameDev for the GameDev vault.

Click the Settings Cog Wheel

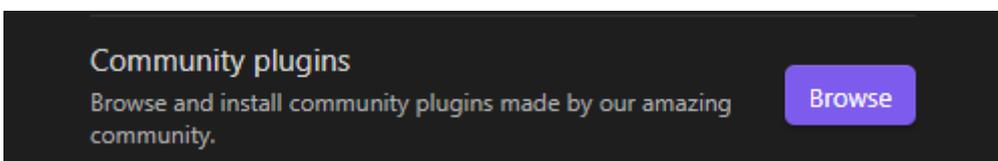


Click Community Plugins

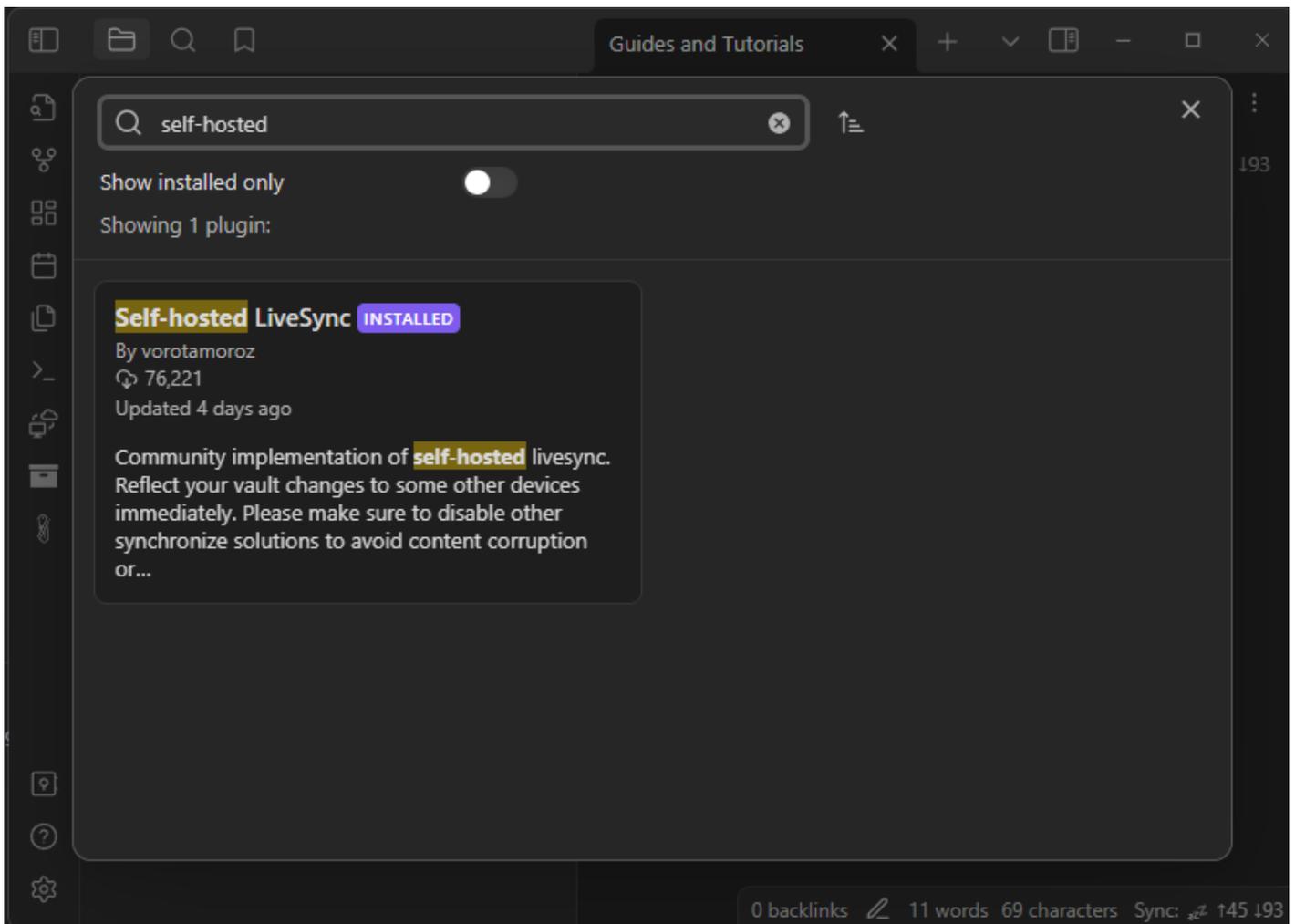


Enable Community Plugins.

Click Browse



Search for Self-Hosted

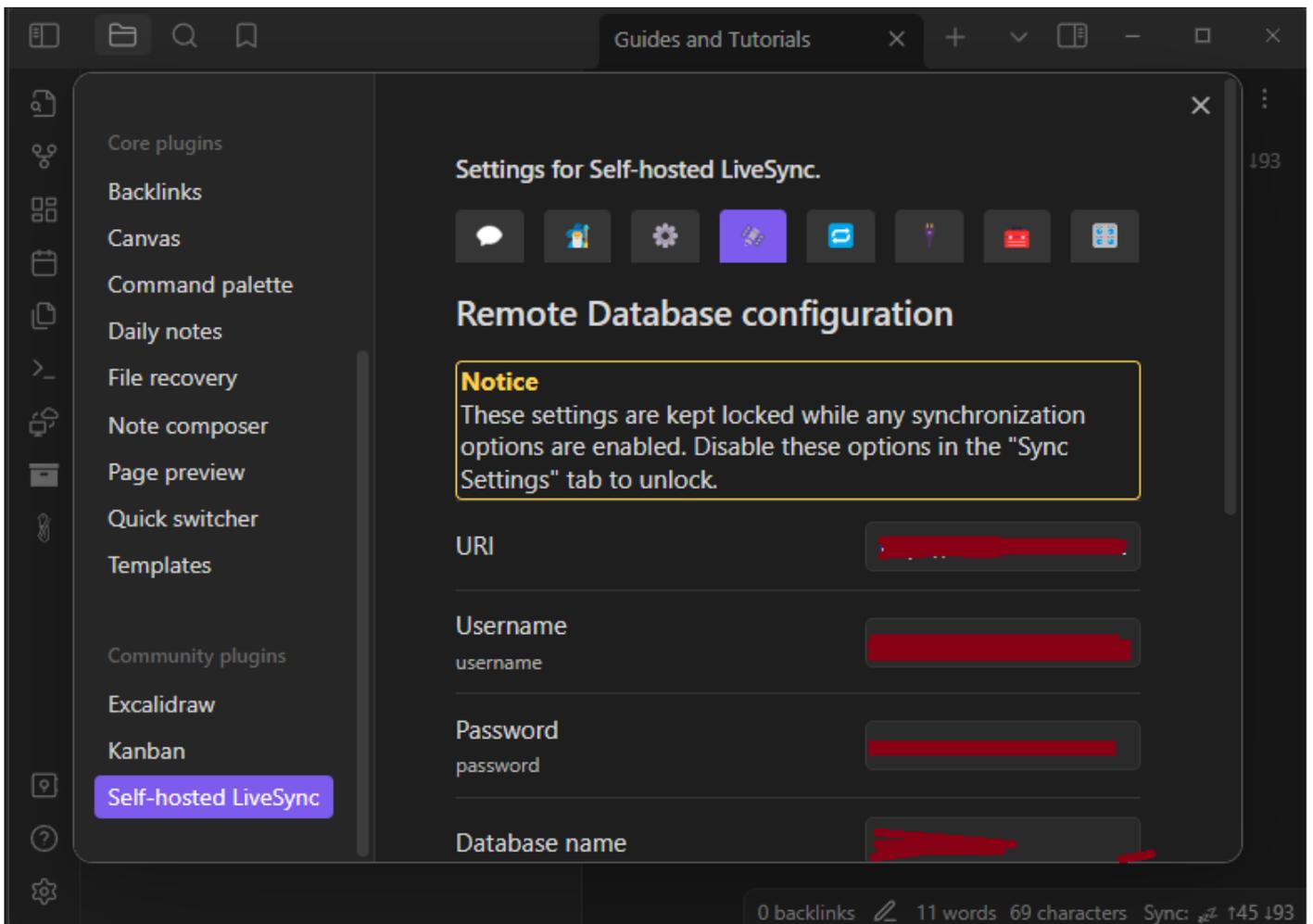


Download and Install Self-hosted LiveSync

ENABLE the plugin.

Go to Settings / Community Plugins / Self-hosted LiveSync

Go to the 4th tab, aka the Remote Database Configuration tab



Enter in the URI, Username, and Password provided by Fifthread, followed by the Database Name

**MAKE SURE YOUR URL DOES NOT INCLUDE CAPITAL LETTERS AT ANY POINT - https needs to be https lowercase!!!**

Click test. Should work.

Go to the Sync Settings tab

Click on the LiveSync Preset and click Apply. You are now auto-syncing!

Guides and Tutorials

Core plugins

- Backlinks
- Canvas
- Command palette
- Daily notes
- File recovery
- Note composer
- Page preview
- Quick switcher
- Templates

Community plugins

- Excalidraw
- Kanban
- Self-hosted LiveSync**

### Settings for Self-hosted LiveSync.

Sync Settings

Presets  
Apply preset configuration

Synchronization Method

Sync Mode

Deletions propagation

Use the trash bin  
Do not delete files that are deleted in remote, just move to trash.

0 backlinks 11 words 69 characters Sync: 145 193