Yet Another Obsidian Git Tutorial (Desktop PC & iPad Sync)

janachoo Sep 17

Yet Another Obsidian Git Tutorial (Desktop & iPad Sync)

Originally posted on Gist

But why?

Well, I haven't seen a single tutorial that covers all the steps I took to achieve a *working* sync solution. There are immensely helpful ones out there, but I still need to hack some of the bits to make it work for me.

As a Git noob, trying to find the said "hacks" from multiple sites and pages was extremely frustrating. I don't trust my fuzzy brain to remember all that either, so I must take note of everything. So, take note I did with this tutorial.

My devices are as follows:

- Desktop: Toshiba Laptop running Bunsenlabs (Debian)
- Mobile: iPad 8th Generation

(As a disclaimer, I have no idea on how to write a tutorial properly. Expect a mess, I apologize in advance.)

Setting Up Git On Desktop

Apps I used:

- Git (see installation docs such as **this** for more info)
- Github Desktop (I installed mine from Flathub… Windows and Mac users can check out the download instructions **here** instead)
- Text Editor
- Obsidian desktop app with Obsidian Git plugin installed

Getting Started - Initial Steps

To get started, I follow the instructions— specifically sections one (Install backup requirements), two (Create Repository), and three (Obsidian Git Setup)— on Danny Hatcher's beginner **tutorial** .

I also add a .gitignore text file on my vault folder (not in .git folder) that contains the line `.obsidian/`.

You can stop at that if you're fine pushing and pulling changes via Github Desktop app. Obviously, I'm **not** (fine), so I decided to configure Git SSH. I chose this method because the default HTTPS option doesn't work on my end.

Configuring Git SSH - All That Remains

Here's what I did to configure Git SSH. First, I follow the instructions on Github Docs about generating a new SSH key and adding it to ssh-agent (saw the link on one of Steve Griffith's (aka Prof3ssor St3ve) videos):

Instructions on generating a new SSH key:

1. Open the terminal and type `ssh-keygen -t ed25519 -C "github_email"`, where `"github_email"` is the email you use on your Github repository.
2. If it was successfully generated, you should see this message: `Generating public/private ALGORITHM key pair.`
3. Then, there would be a prompt along these lines: `Enter a file in which to save the key (/home/jan/.ssh/ed_25519)`, where `/home/jan` is my `/home/user` directory and `/ed_25519` is the algorithm name. I leave it at default, so I simply hit the Enter button to continue.
4. Another prompt will appear: `Enter passphrase (empty for no passphrase):`, where passphrase is the "password" for your SSH key. Type it out, and hit Enter.
5. Type the passphrase again when the prompt `Enter same passphrase again: [Type passphrase again]` appears, then hit Enter.
6. See the confirmation with matching fingerprint graphics on the terminal? If yes, then congrats, you just generated a SSH key!

Now, onto adding the key to ssh-agent:

1. Open the terminal and switch to root. In my case, I ran `sudo -s -H`.
2. Start the ssh-agent in the background by typing `eval "$(ssh-agent -s)"`
3. Finally, add the key to ssh-agent by typing `ssh-add /home/jan/.ssh/id_ed25519`, where `/home/jan` is my `/home/user` directory.

Next, I added the keys on my Github account using the instructions here:

1. To see the key, I typed `cat /home/jan/.ssh/id_ed25519.pub` on the terminal, where (for the nth time) `/home/jan` is my `home/user` directory (Note: If you save your key elsewhere, change the directory accordingly). Hit enter, and copy the key.
2. On Github website, log in your account, click your profile photo, then click "settings".
3. On the settings page, click "SSH and GPG keys" (look at the left sidebar).
4. Click "Add SSH key" (or "New SSH key" if you already have existing keys).
    - For the "Title" field, adding a descriptive label is recommended. In my case, I typed `Toshiba Laptop (Bunsenlabs)`.
    - In the "Key" field, paste the key you copied from the terminal.
    - Click "Add SSH key" to confirm changes.

Another set of instructions down, one more to go.

So, the last step is to set your .gitconfig in your vault to use SSH:

1. On Github website, access your repository page. You know you're right there when the URL is `https://github.com/github_username/repository_name`.
2. Click "<> Code" button and under Local>Clone, switch from HTTPS to SSH. Copy the link that looks like this: `git@github.com:github_username/repository_name`
3. Open your Obsidian Vault folder, then the .git folder (tick "show hidden files" in your file explorer if needed).
4. Open the "config" file inside the .git folder with a text editor.
5. Under [remote "origin"], change the default HTTPS URL to the SSH one you copied on step 2 and hit save.

With this, you can push and pull changes on your desktop within the Obsidian app using the Obsidian Git plugin. Just type your SSH key passphrase when prompted.

Setting Up Git On iPad

Apps I used:

- iSH Shell (a free Linux-like shell on iOS)
- Obsidian mobile app (no need to install Obsidian Git plugin)

Preparing iSH

Before you can use iSH properly for syncing your Obsidian vault, you must install Git and busybox (DOAS) first. Type the following commands:

1. `apk update`
2. `apk upgrade`
3. `apk add git` for Git, then `apk add doas` for Busybox

That's it. Onto the next!

Using iSH To Connect Obsidian Mobile To Desktop's Git Repository

The following steps are based from a Gist **tutorial** made by Professor Danny Quah. So many Danny's, ha!

But before proceeding, get a personal access token first since you'll need it in the `git remote add origin` step later.

Here's how to get the said tokens, courtesy of this forum post:

1. On Github website, log in, click your profile photo, and then click "Settings".
2. On the "Settings" page, click "Developer Settings" (look at the left sidebar).
3. Select "Tokens (Classic)" under "Personal Access Tokens" section.
4. Select "Generate New Token", then "Generate New Token (Classic)". Confirm access by typing your Github account password if needed.

5. In the “Note” field, you can input whatever name you want for your token. In my case, I simply typed `For Obsidian Git`. I also set my expiration date to no expiration. Then, under the “Select scopes” section, I checked “repo”.
6. Click “Generate Token” button. On the next page, copy the token before leaving or else, you would need to regenerate it again. Oh, and also, do NOT share it.
7. Paste your token on the part I labeled as `personal_access_token`: `https://personal_access_token@github.com/your_username/repository_name.git`.

There you go. Save this for later (for step 6 below). It should look something like this: `https://ghp_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX@github.com/workingja/repository_name.git`

Now, do this:

1. Open Obsidian Mobile app and create a new local vault (don’t turn on the “Store in iCloud” option). For simplicity, I name it as the title of my Github repository. Close the app after.
2. Open iSH and make a mountpoint by typing `mkdir /mnt/your_username/Obsidian`, where including `/your_username` is optional. I used it, anyway. Danny Quah used “dq” while I used “workingja” as in `mkdir /mnt/workingja/Obsidian`.
3. Assuming you include `/your_username` too, mount the directory by typing `mount -t ios null /mnt/your_username/Obsidian`. In the file picker, navigate to the main Obsidian folder (not the specific vault folder) and select it.
4. Change directory to the vault you created in step 1 by typing `cd /mnt/your_username/Obsidian/vault_name`. Hit enter.
5. Initialize Git by typing `git init`.
6. Add the Git remote by typing `git remote add origin https://personal_access_token@github.com/your_username/repository_name.git`.
7. Assuming you didn’t rename your Github branches, type `git branch -M main`. Hit enter.
8. Do `git pull --rebase origin main`, hit enter, and wait for it to finish downloading your files.
9. Finally, type `git push -u origin main`, and then hit enter.

You’re done setting up your desktop, mobile, and Github repository for syncing. But of course, it won’t automatically sync like Google Docs. It’s Git, alright! The section below details how to do it (manually).

Workflow To Actually Sync Your Obsidian Vault Across Devices

The steps below are lifted once again from Professor Danny Quah’s tutorial . Credits to him!

Using Obsidian on Mobile:

1. Open iSH. I personally don’t unmount, so I don’t need to type the mount command again. But if you do, type `mount -t ios null /mnt/your_username/Obsidian` and hit

enter.

2. Change directory to your vault using the command used earlier (`cd /mnt/your_username/Obsidian/vault_name`). Then, pull changes with `git pull`.
3. Work away and use Obsidian mobile app. Once you're done, change directory to your vault again (it resets whenever you close the iSH app, use the same cd command as step 2) and commit-push changes by typing `git add .; git commit -m "Message"; git push`.

Using Obsidian on Desktop:

1. In Obsidian desktop app, open the command palette and select "Obsidian Git: Pull". You can also do it via Source Control sidebar tab which you can open via command palette>"Obsidian Git: Open source control view".
2. Work away and use Obsidian desktop app. Once you're done, commit-push changes by selecting the following on the command palette: "Obsidian Git: Create backup" OR "Obsidian Git: Commit all changes", then "Obsidian Git: Push". Alternatively, you can use the Source Control sidebar tab just like in step one.

Credits

Big thanks to:

- Danny Hatcher
- Steve Griffith
- Danny Quah
- Fleeting Notes Developer Matthew

Without their videos and write-ups, this tutorial of mine wouldn't be possible.

Created by Jan (workingja), 2023-09-17

Iphone Obsidian Git Pull Broken, nothing has changed between today and yesterday

---

christer.johansson Oct 9

You can also have separate folders synced across devices by using GIT sparse-checkout approach. On each device you can set what folders it will sync with main repository. Thus, you can keep work related notes updated in your main vault and on your work compputer without sharing personal notes on your work machine. Same approach can be used on smartphones, if you can get GIT working on that device.

I sync my phone in my selfhosted cloud solution with Obsidian app.

---

dubietbay Oct 28

use mini a-shell instead of ish it will be faster use mini a-shell instead of ish it will be faster

New & Unread Topics

Topic	Replies	Views	Activity
How I keep my Obsidian vault super organized with the “Auto Note Mover” plugin	0	569	Jan 23
Anyone want to share/showcase their Canvas creations? canvas	4	2.8k	Dec '22
Olivier’s Theme — a sophisticated, colorful, typographically refined theme	5	1.4k	Jul 28
Demystifying Obsidian’s Outlining Superpowers	1	486	Apr 20
A Plugin for Defining Custom State of Task Items!	2	402	May 9

Want to read more? Browse other topics in or view latest topics.