

Git starting guide 2: Work & manage repositories

Created by: Robert Gan. Last update: Aug. 31st, 2021

Working and managing your local and remote repositories can feel overwhelming. This guide is here to make it as easy and as straightforward as possible. I would recommend reading Git starting guide 1: Initial set up to make sure you have the correct setup before starting to work on your files.



In this second article, I will give you a detailed tour on how to manage your repositories and some basic working tips of git on your computer.

1. Basic working tips

1. **Work on your local repository**
2. **Synchronize to your remote repository**
3. **Bonus: Git ignore - synchronize only the files that matter**

2. Manage your repositories

1. **Timing to create a repository**
2. **Rename & move repository**
3. **Manage your remote repositories in GitHub**

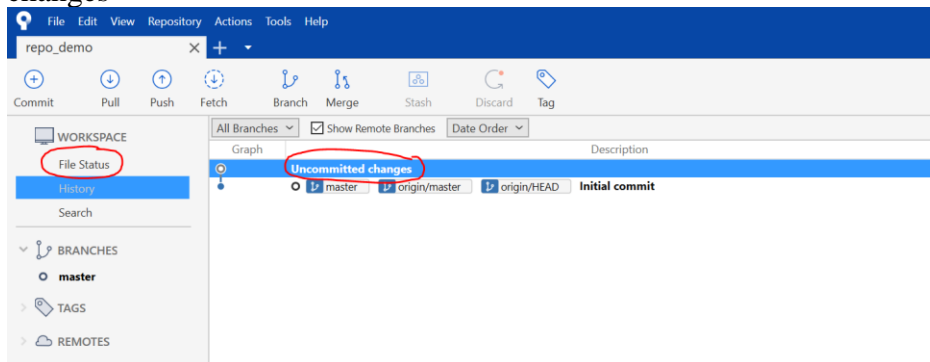
1. Basic working tips

a. Work on your local repository

- Starting from now, you can just work as usual, create files and copy paste files into the folder you created (your local repository)
- You made any changes to your folder (changed one code file, or imported/deleted any files from that folder), and you are happy about the current state of that folder.
- Save your files locally.

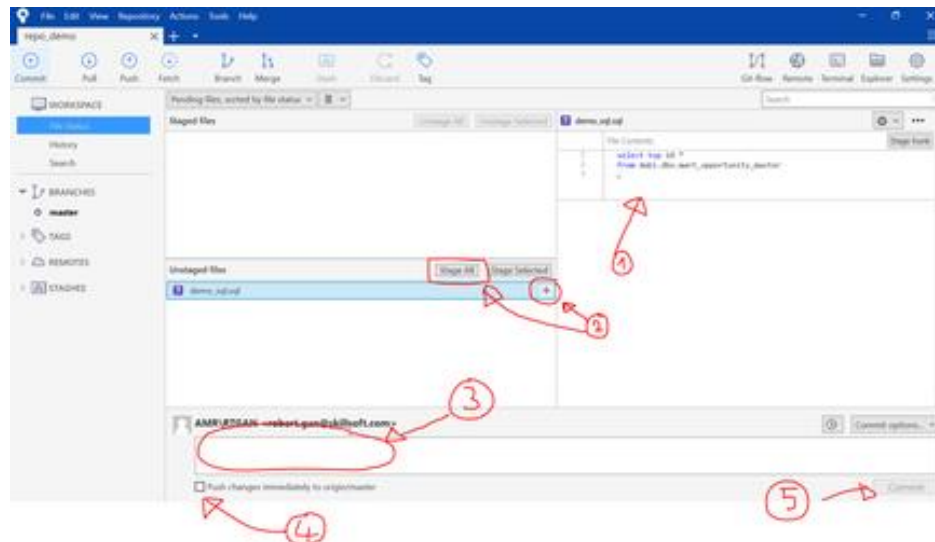
b. Synchronize to your remote repository

- Now that you are happy with the state of the files, you should check those files into git/synchronize the files into the remote repository.
- Go to Sourcetree, and you will see that sourcetree says that you have uncommitted changes

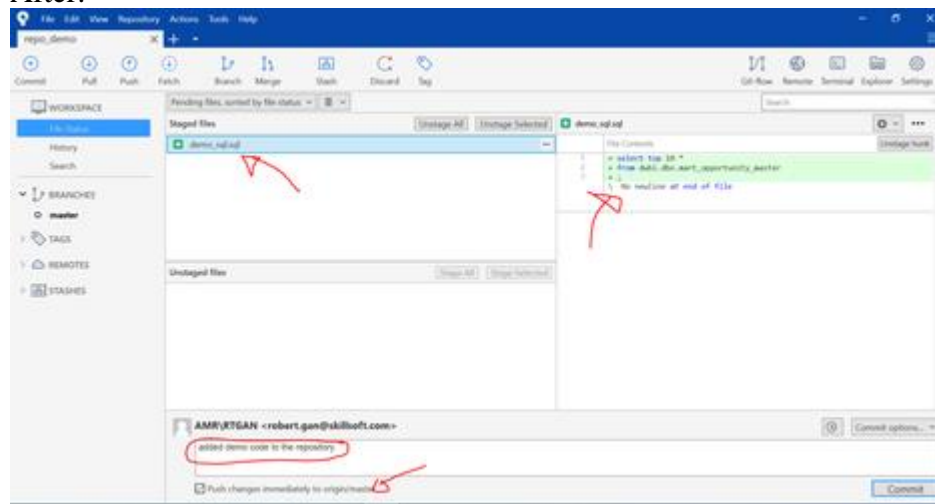


- Go to “File Status”, to choose from the changes that were made
 - 1) you can visualize the changes that were made to your files
 - 2) you can stage specific files or stage all files
 - 3) You can add comments/notes to your commit, so that you can remember why and how you made the changes
 - 4) In case you are the only person working on these files, you can check the box: “push changes immediately to master branch”. If you are collaborating with other people you probably should not do that. Instead, you should look into the details of the branching strategies. (I will have other confluence pages on branches and branching strategies)

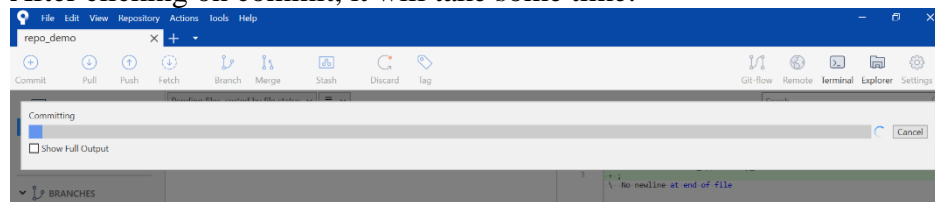
- 5) If you are happy with what you had in the previous steps, you can click on commit on the bottom right corner.
- Before:



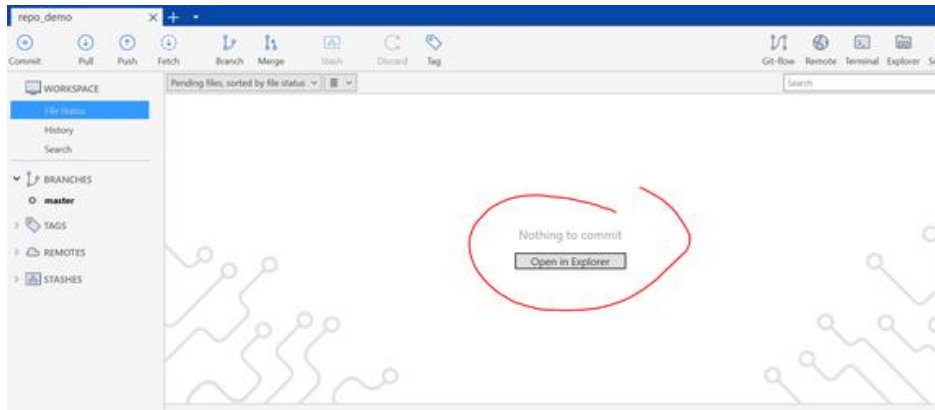
- After:



- After clicking on commit, it will take some time:



- After a couple seconds, this following screen will prompt, and it means that you have successfully synced your local repository and your remote repository. Congrats!



In short, the above can be summarized in the following 5 steps:

1. We create remote repo
2. We clone that remote repo to a local folder, that local folder is the local repo
3. We make changes to the local repo, we saved our files locally.
4. We used sourcetree to sync all the local changes to the remote repo.
5. Now you can repeat the steps 3 and 4, make changes, save them and synchronize the changes to the remote repo.

c. Bonus: Git ignore - synchronize only the files that matter

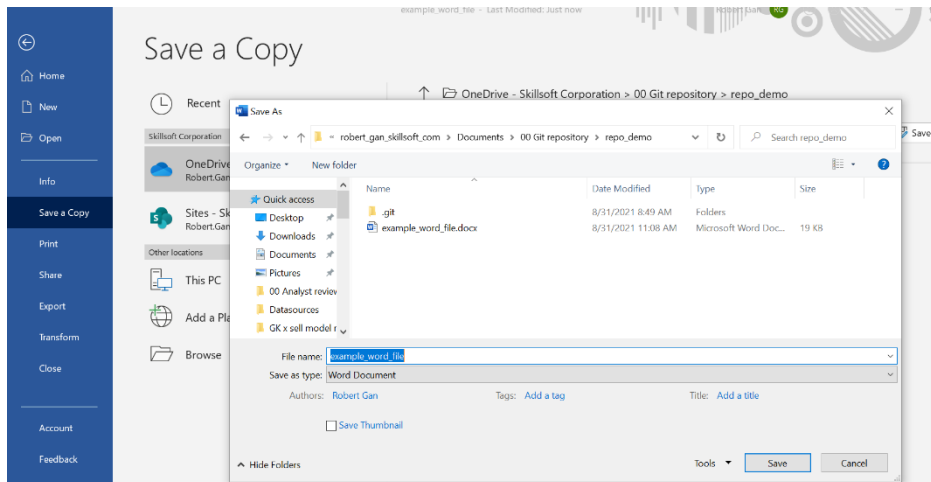
One bonus tip here is on “Git ignore“. You can add a git ignore file to your local repository to ignore synchronizing some files to the remote repository.

A couple use cases:

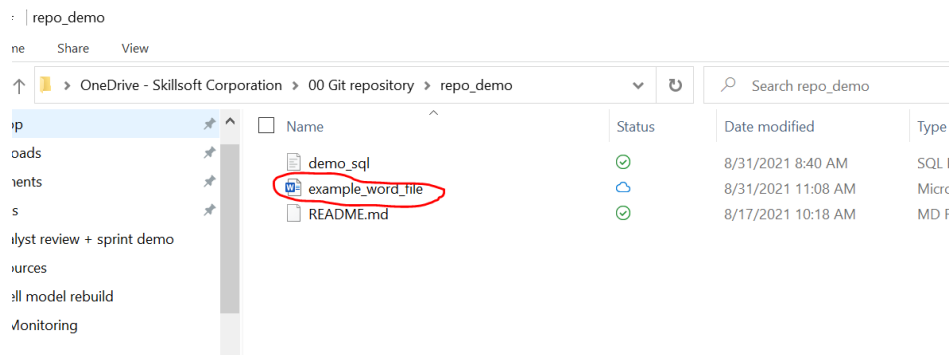
- Heavy Tableau workbooks (.twbx, .twb) that you want to keep in the same folder dedicated to the current project, might not be worth synchronizing to the remote repository.
- Files like presentation(.pptx), word document(.docx) or excel sheets(.csv, .xlsx) that are in the scope of the project but do not need to be backed up in your remote repository.

Below is the walk-through:

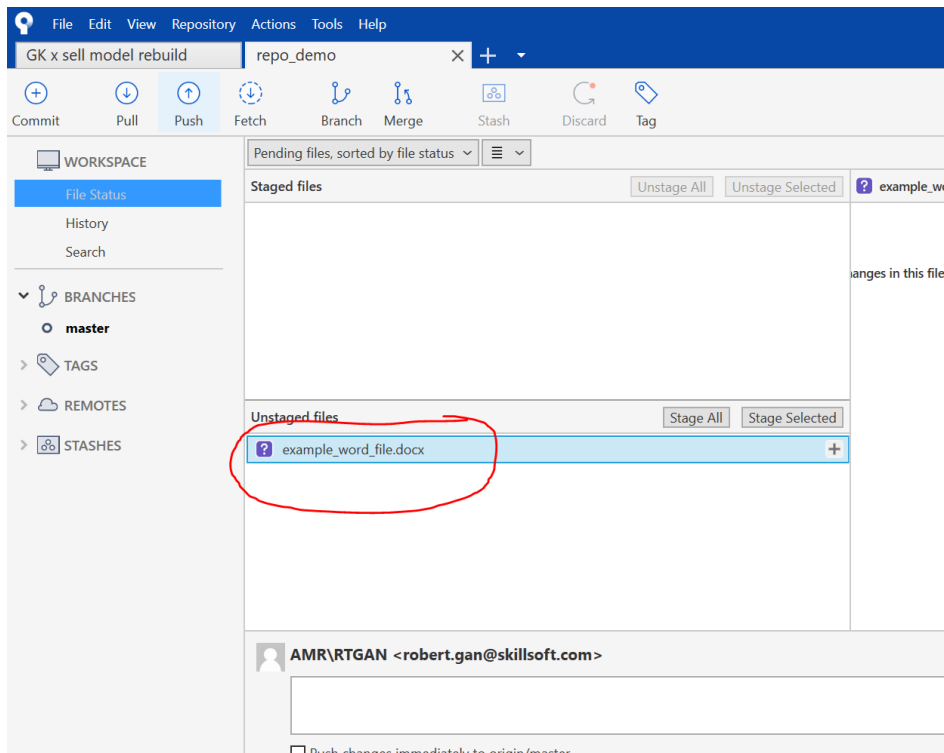
- Let's take the example of a word file.



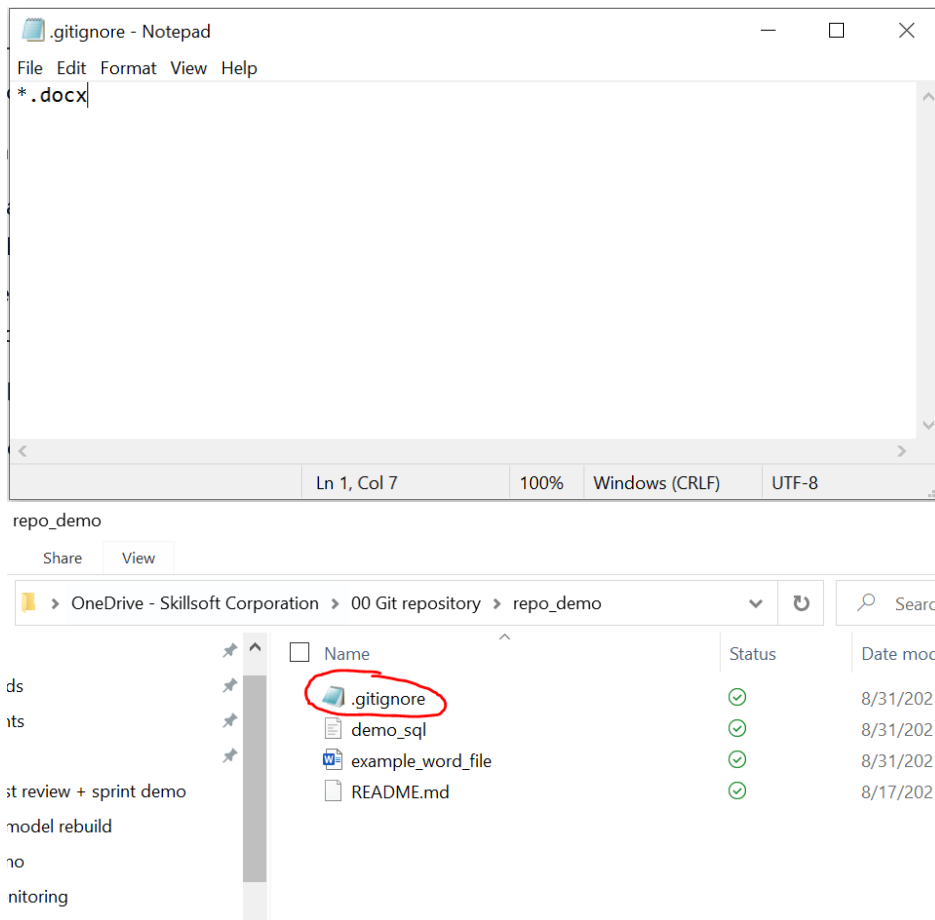
- You added a word file to your local repository



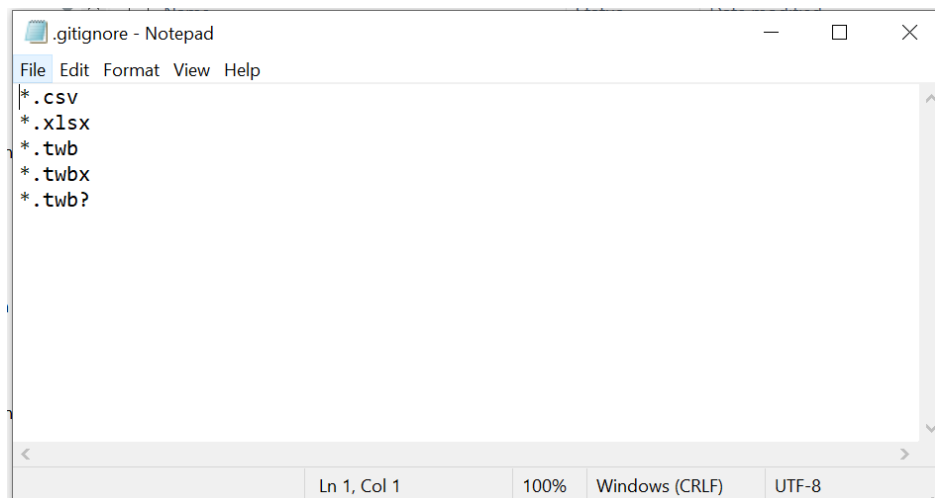
- The message is prompted in sourcetree asking if you want to push it to the remote repository. You don't need to, and you would like sourcetree to ignore this type of files



- Now, in order to ignore the word files, all you need to do is to add a “.gitignore“ to the local repository. open your notepad, add the following lines to the notepad file. Save the txt file as .gitignore to the local repository.

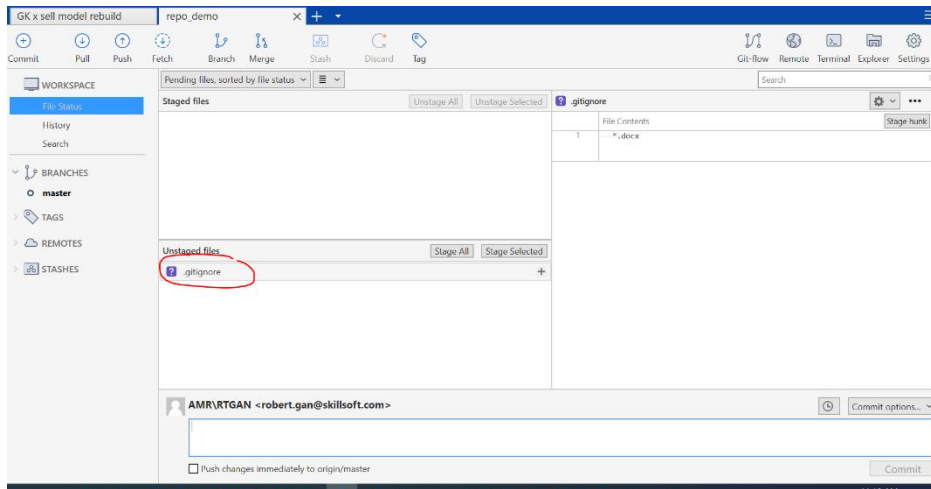


- In the future, for every type of file you do not want to synchronize to your repository, you can just add them to the gitignore txt file



- After saving this file to the local repository folder, go back to sourcetree, and you will see that there's no more messages prompted for synchronization of the word files anymore. Instead, Sourcetree is asking you to synchronize the gitignore file, which is a very light

.txt file that will help Sourcetree understand all types of files that you do not want to synchronize.



2. Manage your repositories

a. Timing to create a repository

It is optimal to create a repository at the beginning of a project, since it allows you to have a complete history of your codes and the possibility to revert back to any versions from the beginning of the project.

Sometimes, you might forget to do this, but it is a good habit to keep creating the repository consistently at the beginning of projects and to checking working versions into the repository.

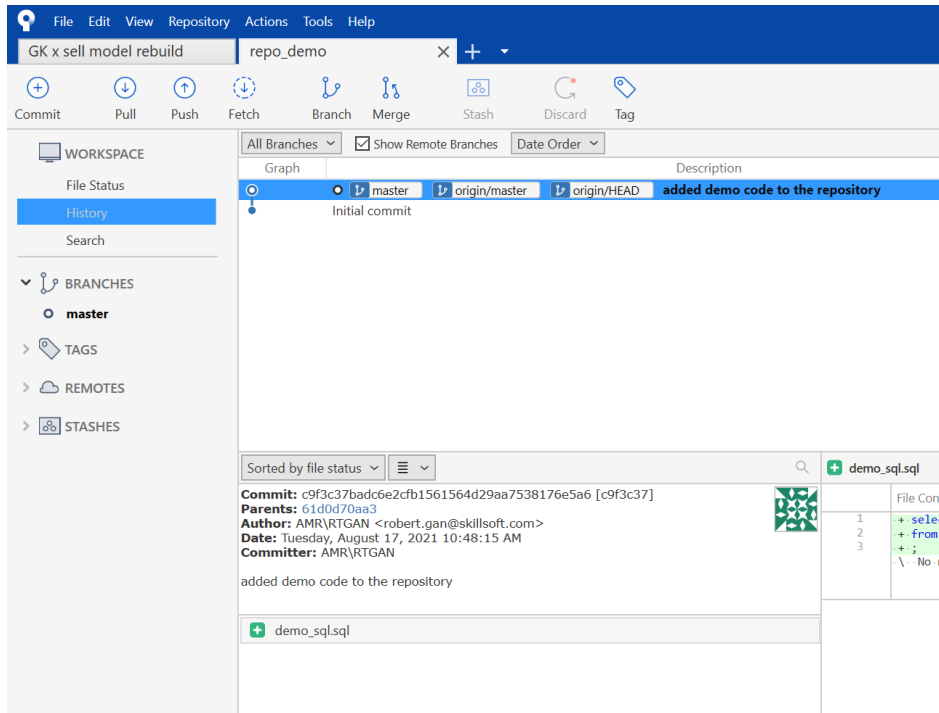
b. Rename or move your local repository

You can rename your local repository the way you want. Your local repository does not need to have the exact same name as your remote repository, but keeping the name aligned can be easier to keep track of your repositories.

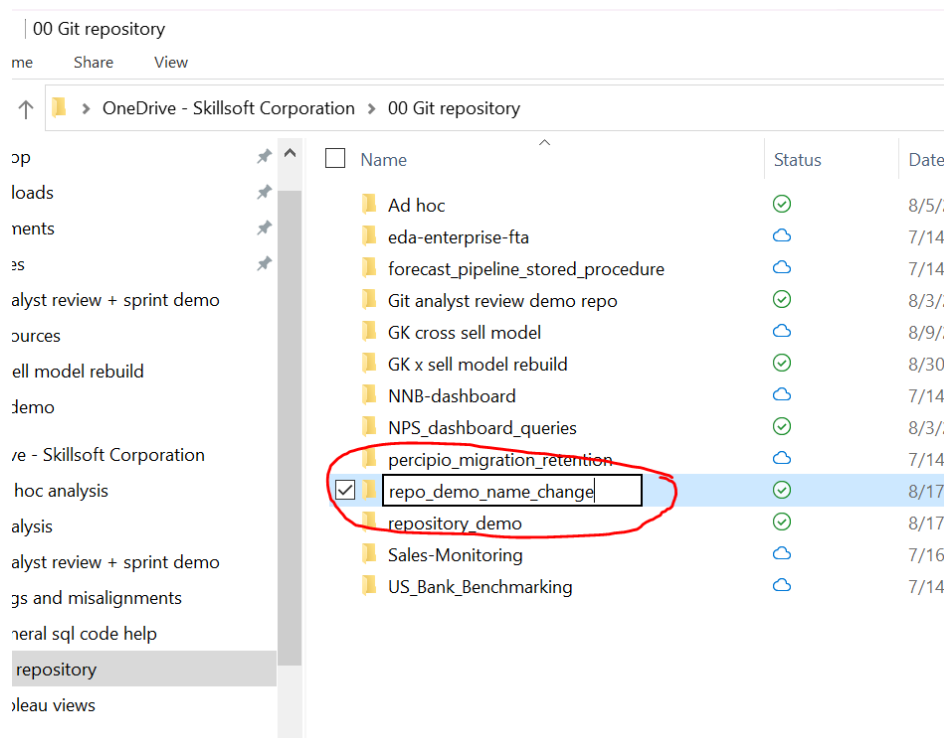
In case you want to rename or move your local repository for whatever reason, you need to be careful. Sourcetree is opening your repository directory specified at the beginning. If you move it or change it, you will need to correct it in order for Sourcetree to work correctly.

I'll walk you through the steps below:

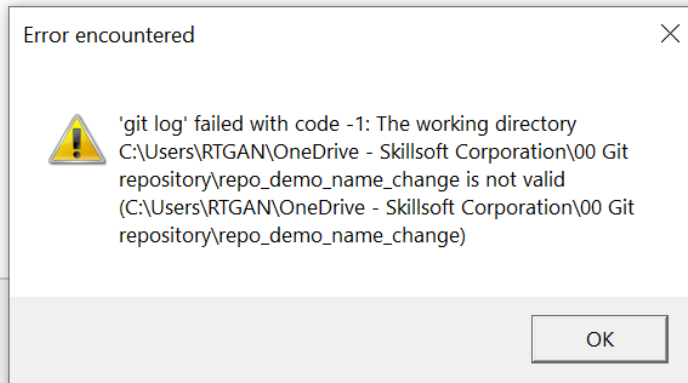
- Check that your repository is opened in a tab



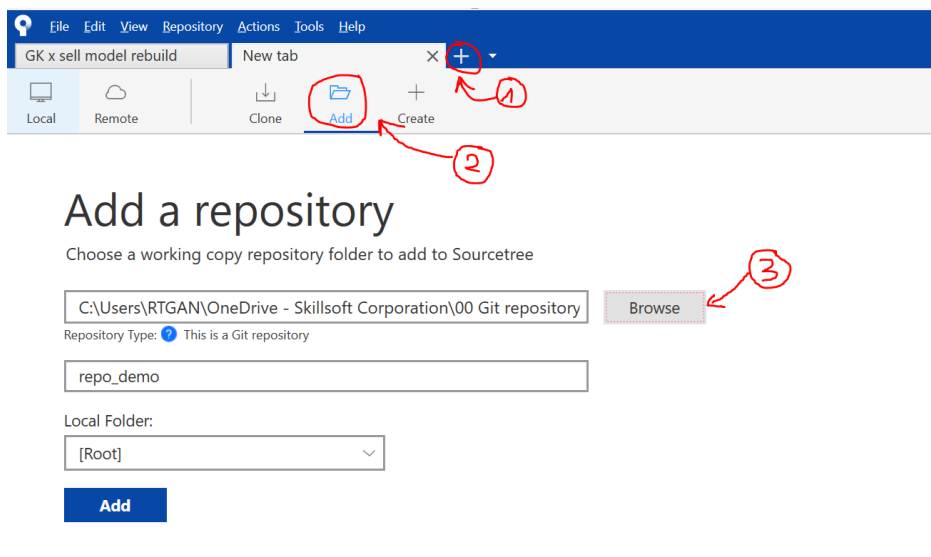
- Change the name of your local repository folder:



- Go back to Sourcetree and go to the repository tab, an error message will prompt:

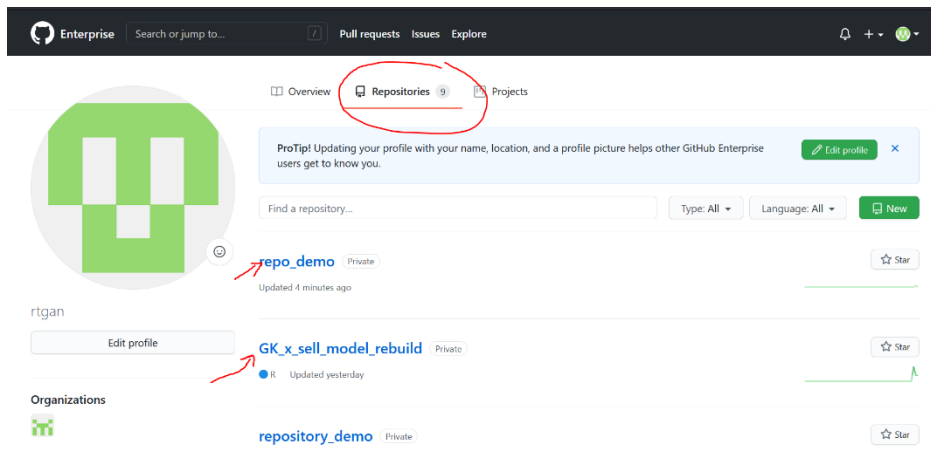


- This message is due to the fact that you changed the specified path of your repository by changing the name. The same message will occur if you move the folder to somewhere else.
- In order to solve this problem, all you will need to do is to re-add the local repository with the Add function, and to specify the correct pathway.

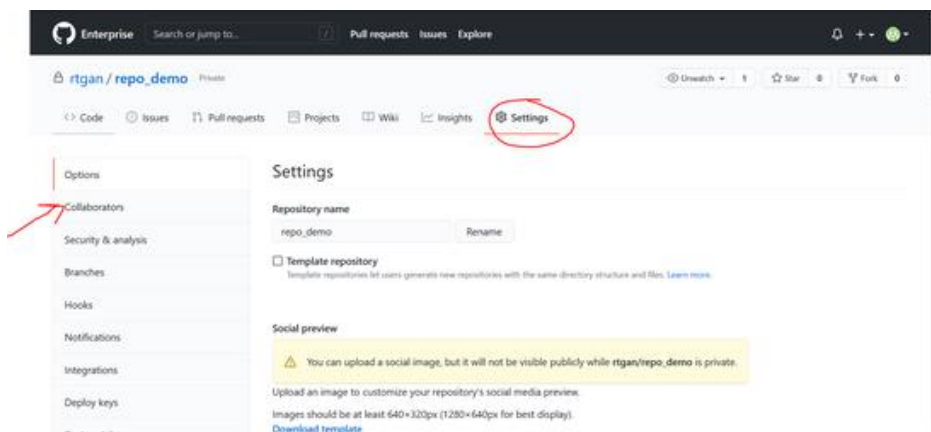


c. Manage your remote repositories in GitHub

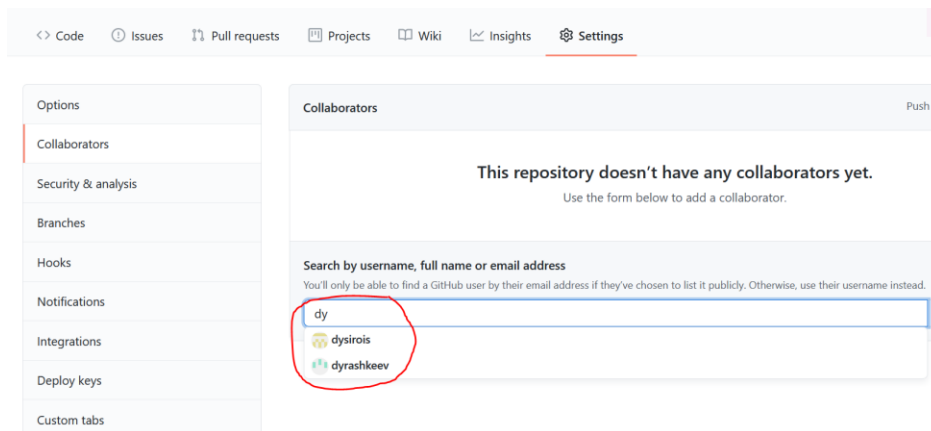
You can see all your remote repositories on the github enterprise website. You can see everything you created:



Click into one of your repository, and you can go into settings and add other people as collaborators in your repository



The name structure is the first letter and the last letter of the person's first name, followed by their last name:



You can then share the link to other collaborator and they will be able to clone the remote repository to their computer and create their local repositories.

Enterprise

Search or jump to...

Pull requests

Issues

Explore

rtgan / repo_demoPrivate

Unwatch1Star

<> CodeIssuesPull requestsProjectsWikiInsightsSettings

master1 branch0 tags

rtgan added the git ignore file to the repo

.gitignoreadded the git ignore file to the repo

README.mdInitial commit

demo_sql.sqlAdded a comment on the query and a

README.md

repo_demo

Go to file

Add file

Code

Clone

HTTPSSSHGitHub CLI

https://github.skillsoft.com/rtgan/repo

Use Git or checkout with SVN using the web URL

Open with GitHub Desktop

Download ZIP

About

No description, w
provided.

Readme

Releases

No releases published

Create a new release