The Starter Guide for GitHub™

A 3 minute guide to help you start contributing on any GitHub project.

GitHub is one of the most popular platforms for modern development teams and it's crucial that developers are comfortable with the role it plays.

In this Guide you'll learn why people use GitHub, the two main benefits it provides and also get a cheat sheet of frequently used commands to help you get started.

WHY DO PEOPLE USE GITHUB?

Github is a platform for teams (or individuals) to host their code & collaborate.

This includes everything from software applications, data projects to quite literally anything else that has files.

Without Github, developers would be stuck working in silos with potentially conflicting code.

Not to mention they'd all have a hard time tracking consistent historical changes.

GitHub also ships with many advanced features that can drastically improve a development workflow.

In a workplace that is becoming increasingly more remote, it's now more important than ever that teams can easily work together.

When used properly, **GitHub can make your development drastically more efficient** and allow you to focus on what you care about most - developing.

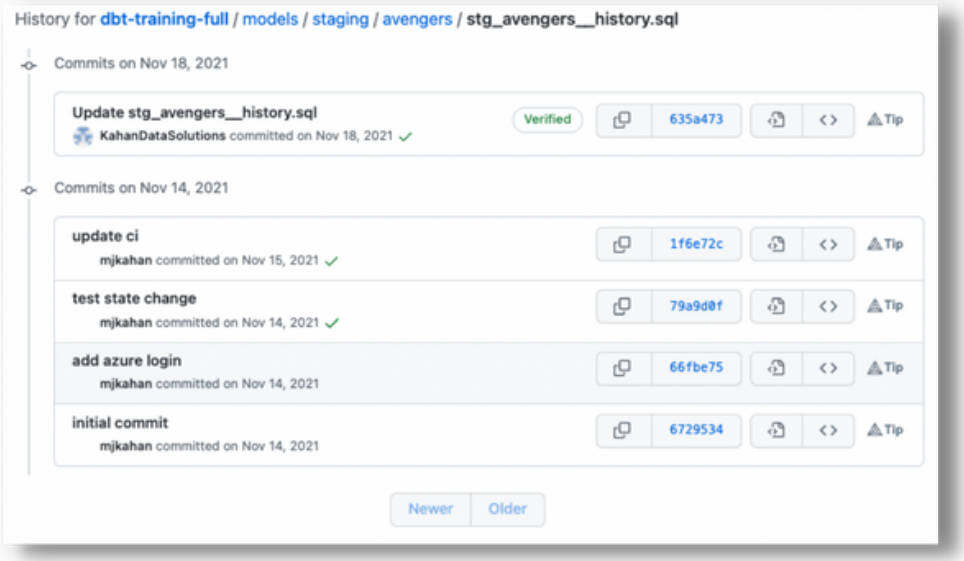
Let's dive into some of main the benefits a bit more.

THE 2 MAIN BENEFITS

There is A LOT you can do with GitHub, but at the top is Version Control & Automation.

Version Control

Version control is the process of keeping track of your code changes over time. Every time a change is made to your project on GitHub, it will keep a record of it and show you exactly what changed. It also allows other developers to easily jump in on a project at any point.



Why You're Better Off Using Version Control:

Rolling Back Changes

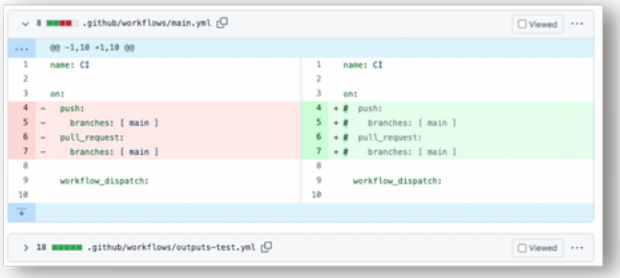
All historical changes are captured and you can easily go back to any previous state of your project

Code Backups

If your computer crashes and you lose all of your files, you can feel confident that the project will still be available on GitHub's servers

Code Reviews

You can work with your teammates to review code changes line by line using GitHub's user-friendly interface.



Conflict Resolution

It's common to accidentally build conflicting code. But GitHub will automatically call out any of these issues so that you can decide what should be kept.

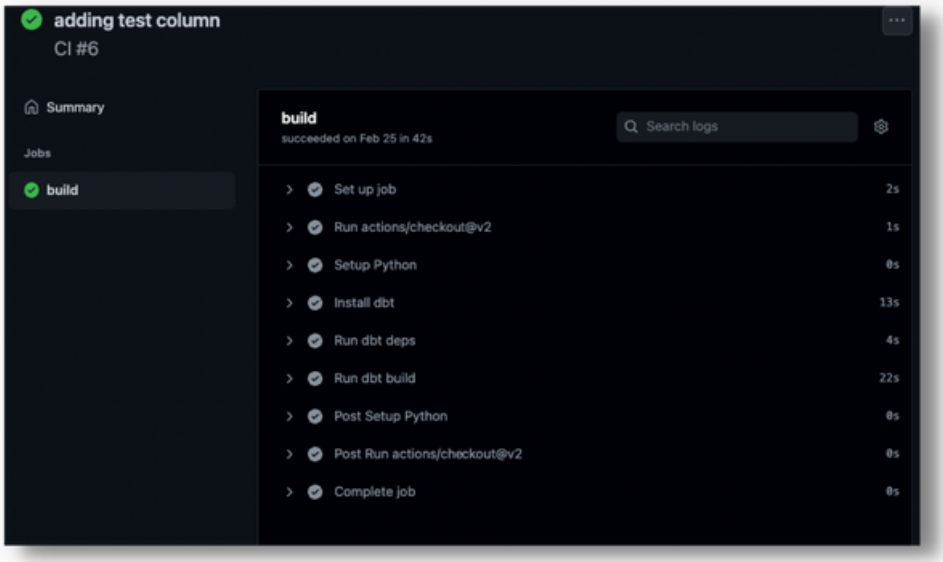
Automation

Automation can take many forms but once you see it work properly, it's hard to go back. One of GitHub's strengths is it's open marketplace which provides a lot of pre-built tools to help you start automating tasks right away.

Here are 3 key ways automation w/ GitHub can help you:

Continuous Testing & Deployment (CI/CD)

Use GitHub Actions to automatically execute commands based on a trigger *(ex. on a merge, commit, etc.)*



A common scenario is to run workflows on a Pull Request .

This helps ensure changes are properly tested and any obvious errors are caught before it gets too far.

Releases

With testing automatically covered, you can feel confident releasing your code to production as needed. This means moving away from scheduled deployments with large time gaps in between to a more continuous deployment process.

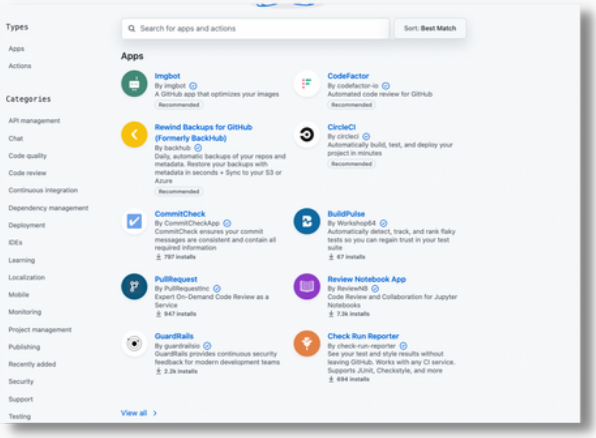
Pro Tip: Create a workflow to automatically compile and snapshot your code as a Release.

Integrations / Notifications

As mentioned, GitHub is known for being developer-friendly and having a great marketplace.

That means there are a ton of ways to integrate your workflows with other tools in your stack and avoid re-inventing the wheel on common tasks.

This alone is a huge reason why people love GitHub.



COMMAND CHEAT SHEET

There are many commands you can run with GitHub, but here are some of the most common ones:

git clone [url]
Get a copy of a repository on GitHub on your local machine

git pull
Get the most recent changes of a repository copied to your local machine

git checkout [branch-name]
Switch to another branch and “check it out”, aka set it as your working directory

git branch
List all branches. You will see * next to the currently active branch

git checkout-m [branch-name]
A shortcut to simultaneously create a new branch and check it out

git push origin head
Send all changes on your local branch to the hosted GitHub version

git add [file-name or .]
Add files in their current form to your next commit. Aka “stage” your changes.

git status
Show modified files in working directory, staged for your next commit

git commit -m “[descriptive message]”
Commit staged files to your branch. All staged changes will be part of the same commit.

For more Syntax details, check out the GitHub documentation here or from the official Git site here.

*Friendly reminder - **Git is not the same as GitHub.***
***Git** is a distributed version control tool that can manage a development project's source code history, while **GitHub** is a cloud based platform built around the Git tool.*

Git is a tool a developer installs locally on their computer, while GitHub is an online service that stores code pushed to it from computers running the Git tool.

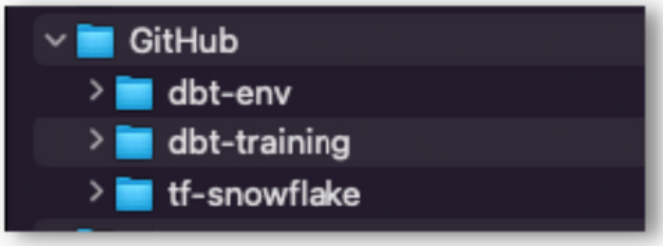
More on this topic here.

BONUS: ORGANIZE YOUR REPOS

Typically you'll be working on multiple repositories at once.

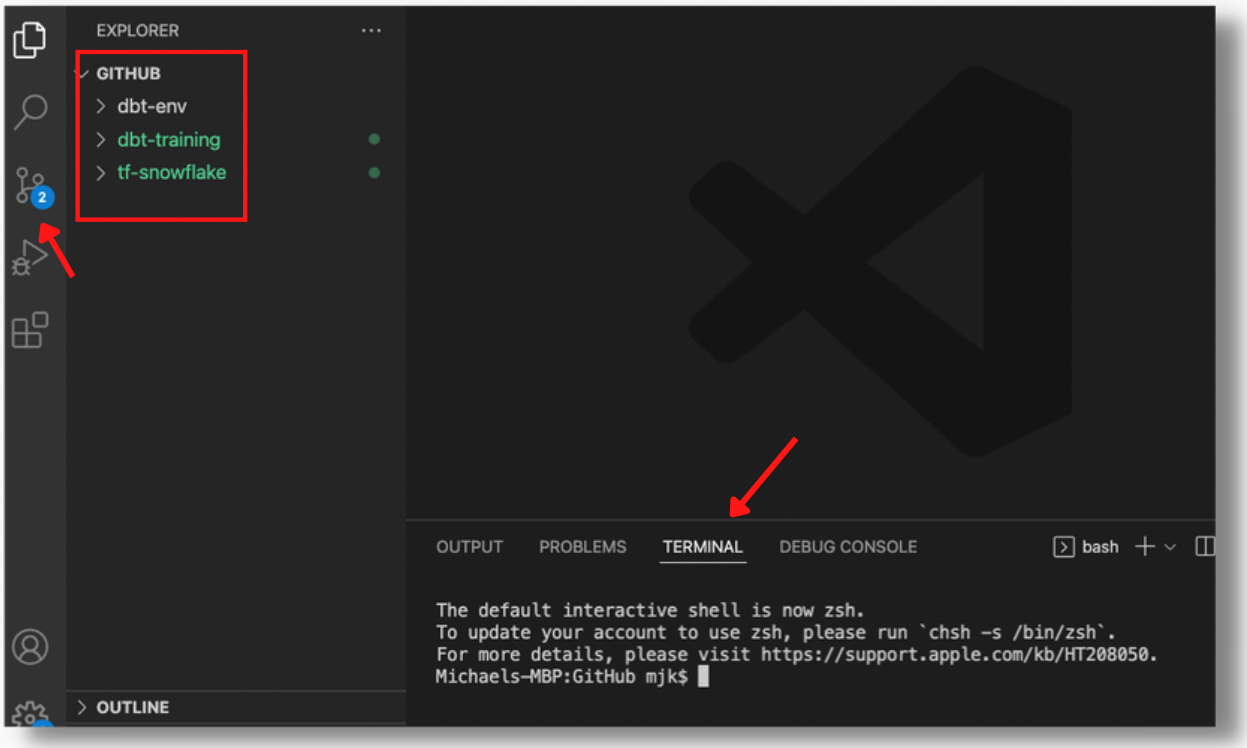
To help keep it all organized, I like to create a single "GitHub" directory on my local computer and clone all projects within it.

It will then look something like this:



Why is this helpful?

Most modern editors (like VS Code) have built in terminals or Git integrations. Keeping your repositories organized this way allows you to easily hop back and forth between your repositories without needing to switch your view.



Thanks For Reading!

I hope you found this guide helpful. There is **so much more** to learn about GitHub, but hopefully now you have a better appreciation for its purpose and the type of things to expect as a developer.

For more GitHub-specific content, check out the rest of my YouTube playlist.

And if you'd like to receive more free tips on modern data engineering, consider joining my newsletter.

Cheers,
Mike

PS - I'm always trying to improve my content. Let me know if you found this Guide helpful (or not) through my Contact page (select option "Content Feedback").