Working with Github

What is Github?

GitHub is a platform for hosting and collaborating on projects. You don't have to worry about losing data on your hard drive or managing a project across multiple computers — sync from anywhere. Most importantly, GitHub is a collaborative and asynchronous workflow for building software better, together.

Sign up for Github

1. Go to https://github.com/
2. Pick a username, enter your email, create a password and Sign up for Github

Github Desktop

GitHub Desktop is the easiest way to get code on GitHub.com.

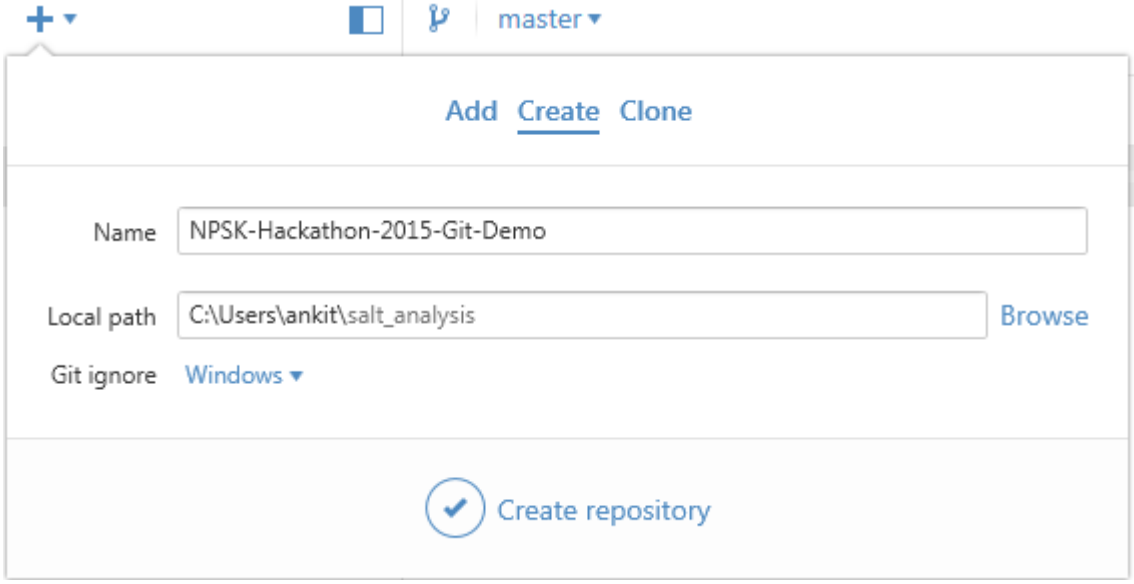
Download GitHub Desktop for **Mac** and **Windows**. Once you install GitHub Desktop, a short set up wizard will walk you through some basic configuration and will help you connect GitHub Desktop with your GitHub.com account that you just created.

Repositories

A **repository** is the basic unit of GitHub, most commonly a single project. Repositories can contain folders and files, including images – anything your project needs. Because we recommend including a README, or a file describing the project, in every repository, GitHub makes it easy to add one at the same time you create your new repository.

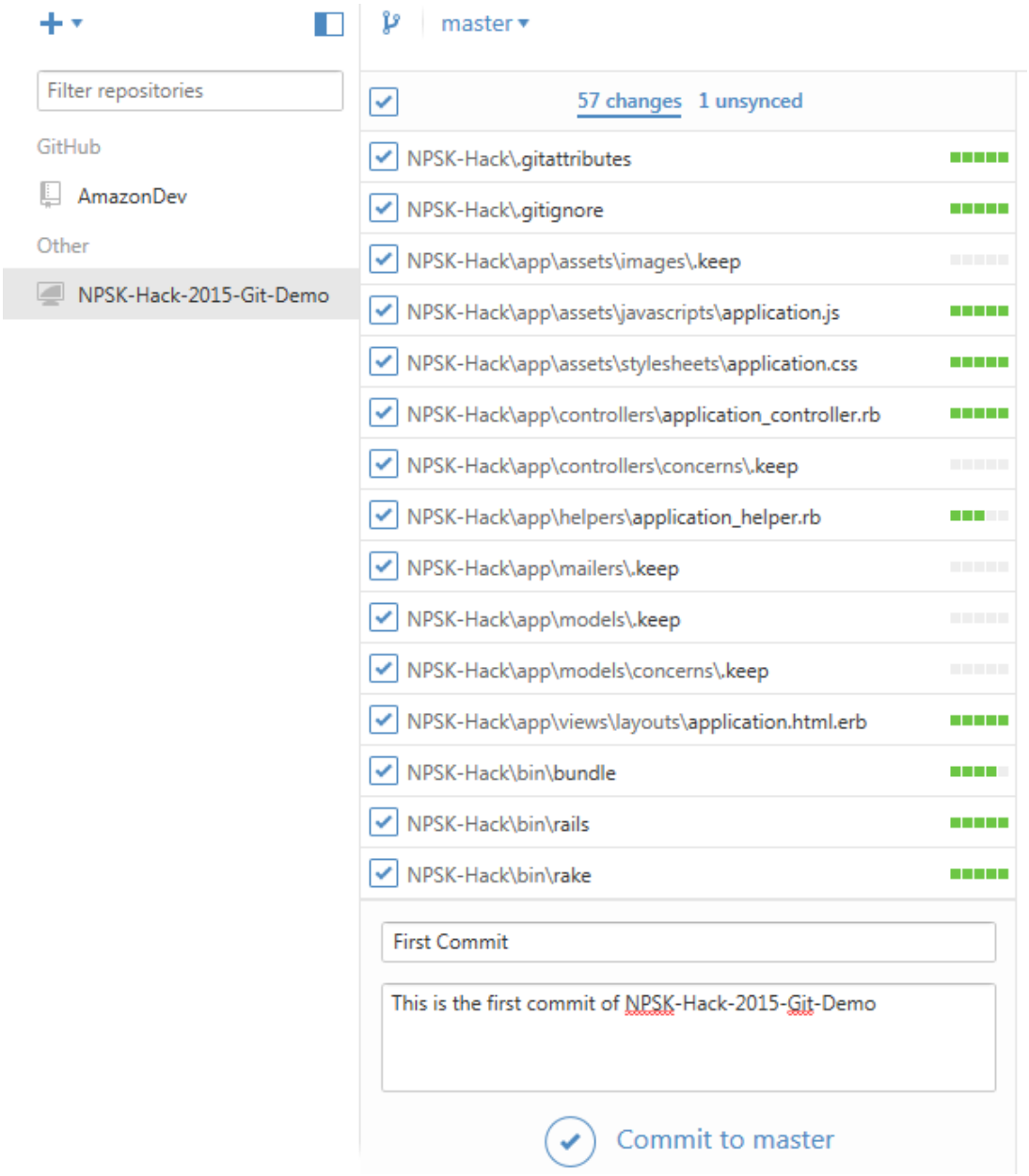
Set up your project in GitHub Desktop

1. Create a repository by clicking the + icon on Github Desktop.
2. Select to **Create a repository** and point to the local path of your project.



3. Enter `NPSK-Hackathon-2015-Your-Hack-Name` as the name of your repository.
4. Click **Create repository**.
5. In File Explorer, create your project files under the same directory as your repository

6. Once you've saved files, add a summary and description of your work so far and click **Commit to <branch name>.**



Commits

On GitHub, saved changes are called **commits**. A bunch of commits together make up the history of your project.

Each commit has an associated **commit message**, which is a description explaining why a particular change was made. Thanks to these messages, you and others can read through commits and understand what you've done and why.

Your first Commit

All Git repositories are based on *commits*—snapshots of your code at a point in time. You need to make at least one commit before you can push your code up to GitHub.com.

Navigate to the **Changes** tab and click **Commit** to create your first commit. You'll need to create a new commit every time you change files. Creating a commit is like saving a file—you are telling Git that you'd like to remember this point in history.

Make as many commits as you like locally. No one but you can see those commits until you push them to GitHub.com.

Pushing code to Github

Click the "Publish" button in the upper-right corner and GitHub Desktop will ask you what kind of repository to create. Choose **Public repository** — anyone can see a public repository, but you choose who can commit (make changes) to it. You can create as many public repositories as you want on GitHub.com for free.

Now that you've published the repository, you have it in two places:

- **Local repository on your computer** — You can work on this repository without an Internet connection using GitHub Desktop. This is where you edit files and make changes to your project.
- **Remote repository on GitHub.com** — You can send people links to your repository on GitHub.com so they can see your code and use all of GitHub's other features (like Issue management and Pull Requests).

Each time you make changes to your local repository, you'll need to sync your changes (by clicking the button in the upper-right corner of GitHub Desktop) to make sure they show up online.

Sync Repository

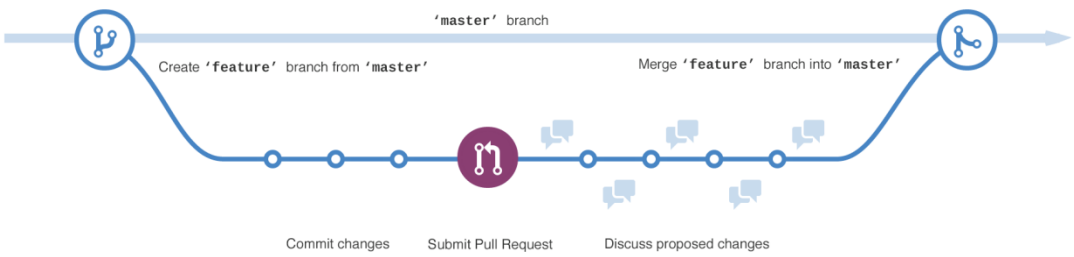
After you've made the first push to GitHub, you'll need to only click the **Sync** button every time you wish to sync committed changes.

Branches

Branching is the way to work on different parts of a repository at one time.

When you create a repository, by default it has one branch with the name `master`. You could keep working on this branch and have only one, that's fine. But if you have another feature or idea you want to work on, you can create another branch, starting from `master`, so that you can leave `master` in its working state.

When you create a branch, you're making a copy of the original branch as it was at that point in time (like a photo snapshot). If the original branch changes while you're working on your new branch, no worries, you can always pull in those updates.

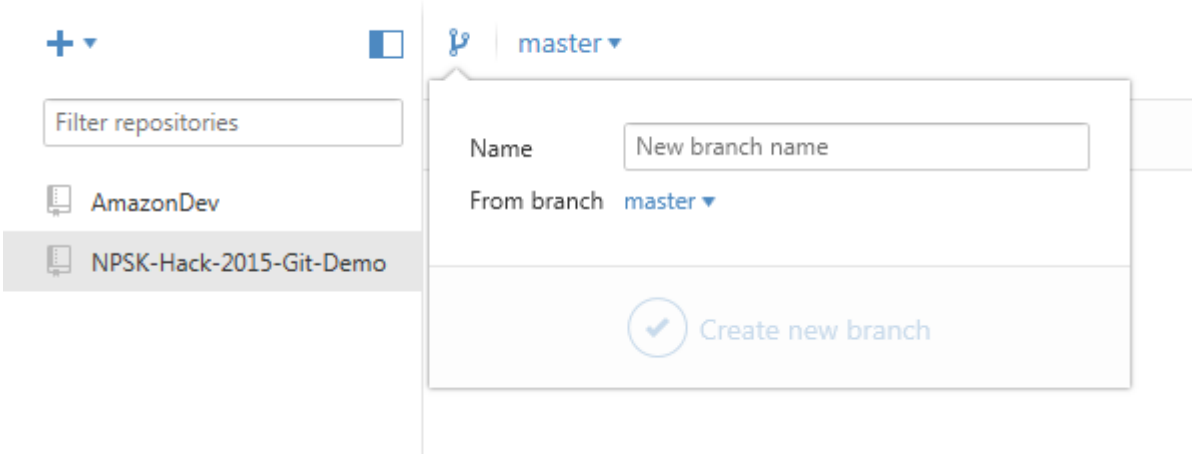


You may have, at some point, saved different versions of a file like, "story.txt", "story-joe-edit.txt", "story-sue-edit.txt". Branches accomplish the same goals but are easier to manage on GitHub repositories.

At GitHub, developers use branches for keeping bug fixes and feature work separate from the `master` (production) branch. When a feature or fix is ready, the branch is merged into `master`.

Creating Branches

1. To create branches, click the **Branch Icon** and enter a name.
2. Click **Create new branch.**
3. To make sure the branch is reflected on GitHub, make sure you sync changes.



4. To manage branches, select the branch name located next to the **Branch Icon**.