Git and Github with Xcode

ITP 342: iOS Development

What we are going to do

In this guide, we are going to learn how to save and update your Xcode projects using Git and Github. This guide provides the basic information you need to get started, and throughout this guide, we will refer to sections in this awesome tutorial by Ray Wenderlich, which gives a better explanation on many details. Let's get started!

Install Xcode from App Store (if you don't have one)

If you already have the latest version of Xcode on your Mac, you can skip this step!

If not, don't worry, installing Xcode on your Mac is not difficult.

1. Launch the App Store app on your Mac, log in using your Apple ID if you haven't.
2. Search for Xcode, and Download it. Enjoy something else while the download is in progress, it may take up to a few hours!
3. Once the download finishes, launch Xcode. Download additional components if Xcode asks you to.
4. When you see a window that says "Welcome to Xcode", you can proceed to the next step!

Creating a GitHub Account (if you don't have one)

If you already have a GitHub account, you can skip this step!

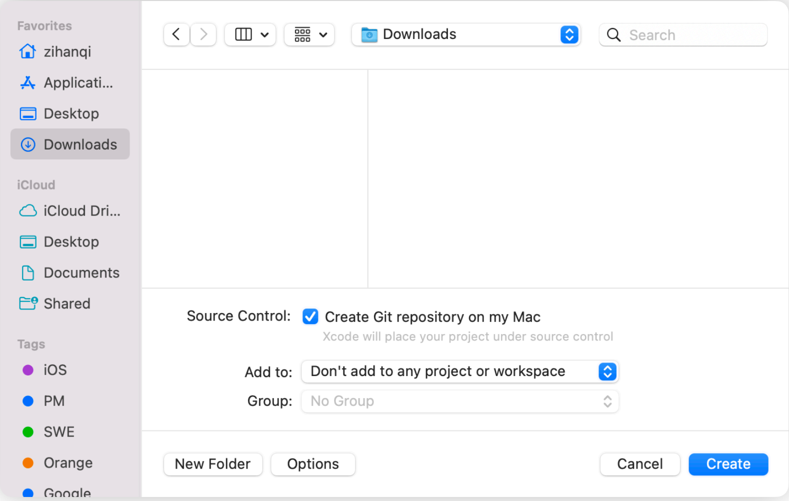
If not, don't worry, setting up a new account is not difficult.

Follow this link to create a new account: https://github.com/join. Using a free account is good enough for this class. After you successfully created a new account, proceed to the next step.

Create a new Xcode project

Create a new Xcode project, give it a good name, like “GitUsePractice”. Keep in mind the following important note:

> **Important**
>
> When creating the new project, you will see a window like the one below.
> Make sure you’ve selected “Source Control: Create Git repository on my Mac.” If you forgot to do this, don’t worry, create another project and repeat this step.



Source Control: ☑ Create Git repository on my Mac
Xcode will place your project under source control

Add to: Don't add to any project or workspace
Group: No Group

Commit some changes in Git

In this step, you will learn how to save changes you’ve made like a pro-developer! Of course, this means that we are going to use Git.

Go ahead and read the “Making some Commit-ments” section in the Ray Wenderlich Tutorial. After you finish, do the following practice:

- Make some new changes to your project (adding code, removing code, adding comments)
- Committing the changes you’ve made to your project in Xcode

Xcode and GitHub

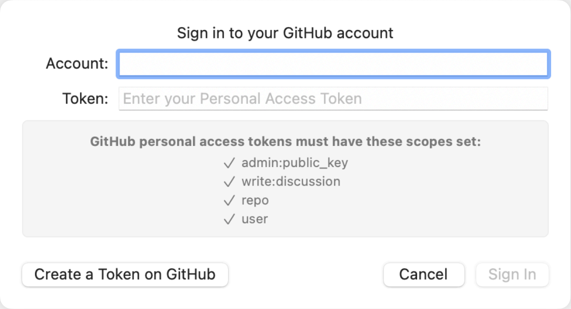
You’ve successfully made some Git commits in Xcode! This is a great start. In real life, you may want to upload your project online for various reasons. Github is one of those places that allows you to store projects online, so that you and others can download at a later time. Once we upload our project to GitHub, we will now have 2 copies of the same project. We often refer to the project uploaded on GitHub as a “remote” copy, and the copy stored on our computer as a “local” copy.

In this step, you will learn how to connect your GitHub account to your Xcode, and create a GitHub remote for our Xcode project.

Before you proceed to read the next section in the tutorial, keep in mind the following:

Important

When adding your GitHub account in Xcode, you will see the window below. Notice it uses a “Token” field, instead of a Password field. Click on "Create a Token on GitHub”, follow its instructions, and use the Token you’ve generated to Sign in here.



Sign in to your GitHub account

Account:

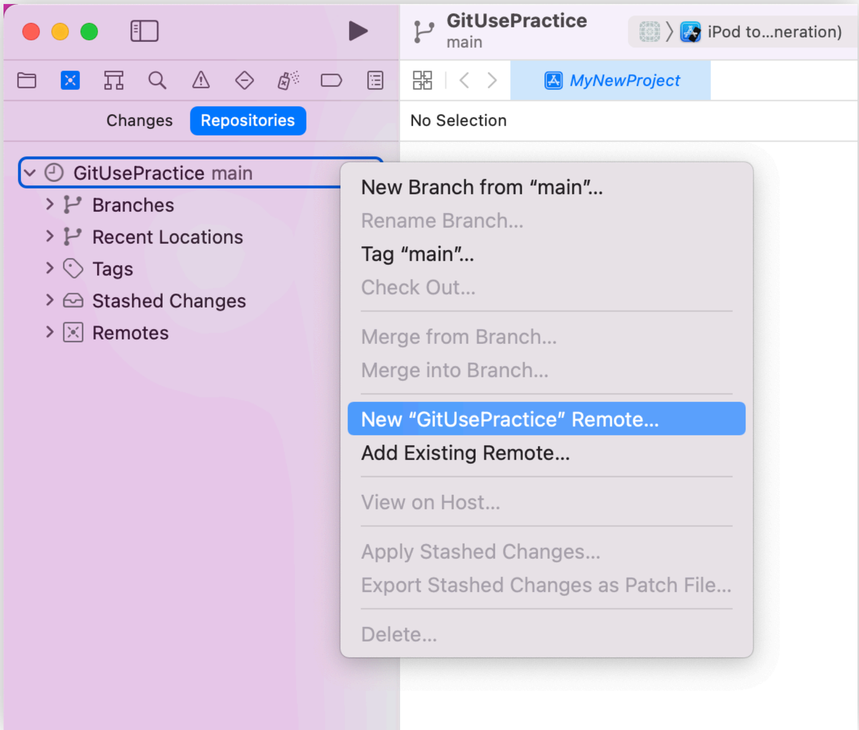
Token: Enter your Personal Access Token

GitHub personal access tokens must have these scopes set:

- ✓ admin:public_key
- ✓ write:discussion
- ✓ repo
- ✓ user

Important

In the tutorial you are going to read, you will read about creating a new remote repository for your Xcode project. In Xcode 13, Xcode will select the “Changes” tab by default. To see the following interface, make sure you select the “Repositories” tab.



Phew! With these important notes in mind, go ahead and read the "Xcode and GitHub" section in the Ray Wenderlich Tutorial.

> Note
>
> You can skip parts in "Xcode and GitHub" section that says anything about branches. Branching allows you to create multiple "copies" or versions of your project. In real life, branching allows different members of a team to work on the same project simultaneously. To learn more, read the "Branching Out" section of the tutorial.

After you finish reading "Xcode and GitHub", do the following practice:

- Create a new GitHub remote for your project.
- View your project on GitHub.

> Experiment
>
> - Make some changes to your project (adding or removing code and comments).
> - Commit these changes using Source Control -> Commit…
> - View your project on GitHub. Can you see these changes? Why or why not?
> - Now, make a Push to your GitHub using Source Control -> Push.
> - View your projects on GitHub. Can you see these changes? Why or why not?

Where to Go From Here?

Congratulations! You have reached the end of this guide, and you have learned all you need to start using Git and GitHub with Xcode. This is a big achievement. You may have been more familiar with using cloud storage and sharing services like iCloud, DropBox, Google Drive, etc, and now you have one more tool in your hand. Great job!

Take a moment to reflect on what you have learned through this guide. What do you like or dislike about using Git? Is there anything that you find frustrating, confusing, or surprising? Why do you think that is? There are many other ways people use Git, including from the Command Line, from software apps like GitHub Desktop, and many other more. If these sound interesting to you, take a moment to explore what’s possible beyond this guide.

> Note
>
> Git and GitHub is widely popular in the industry. For instance, check out apple/swift.

> Note
>
> The Ray Wenderlich Tutorial introduces Git and Github in more details. You can read through other sections to learn more.