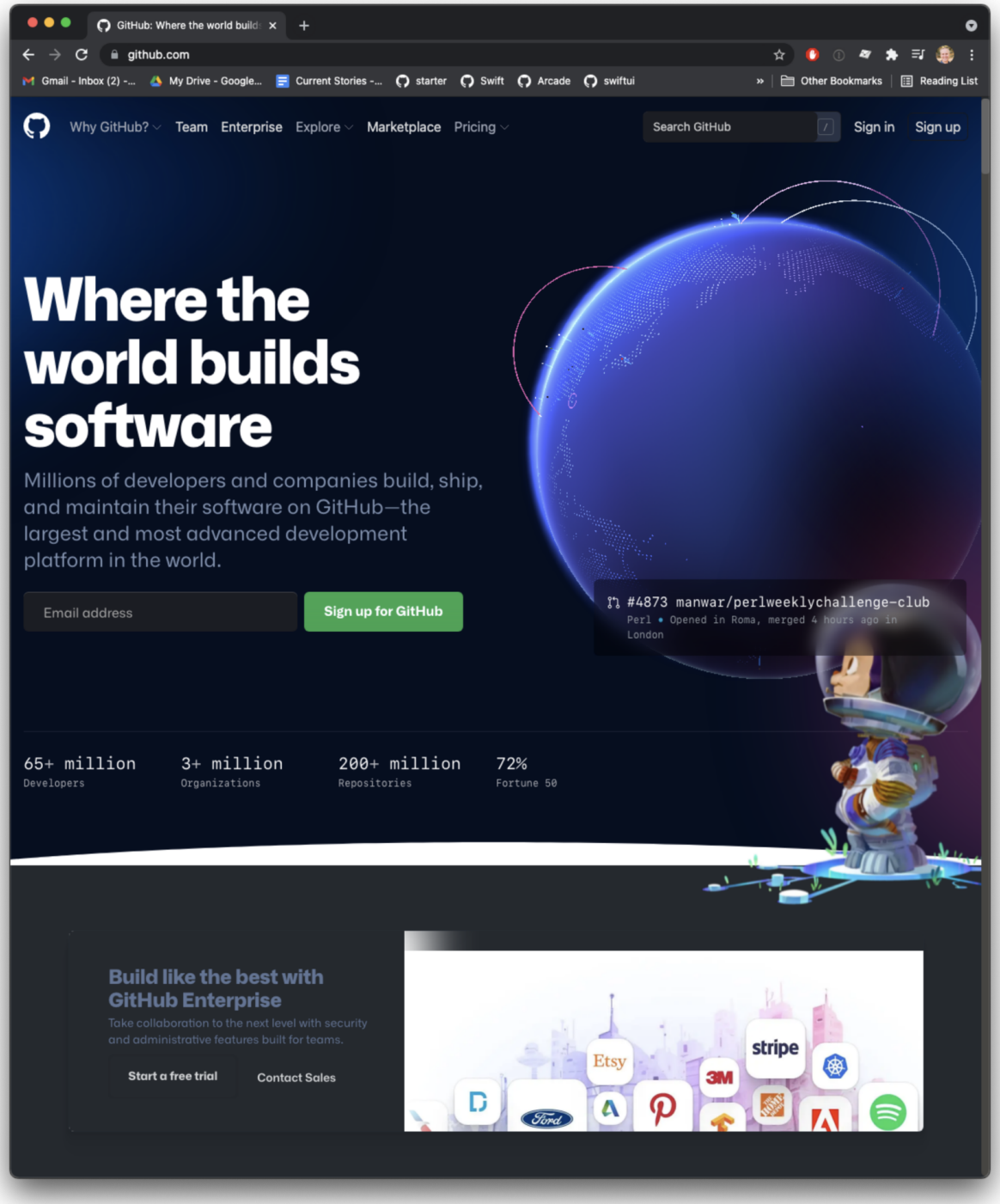
iOS Professional GitHub Setup

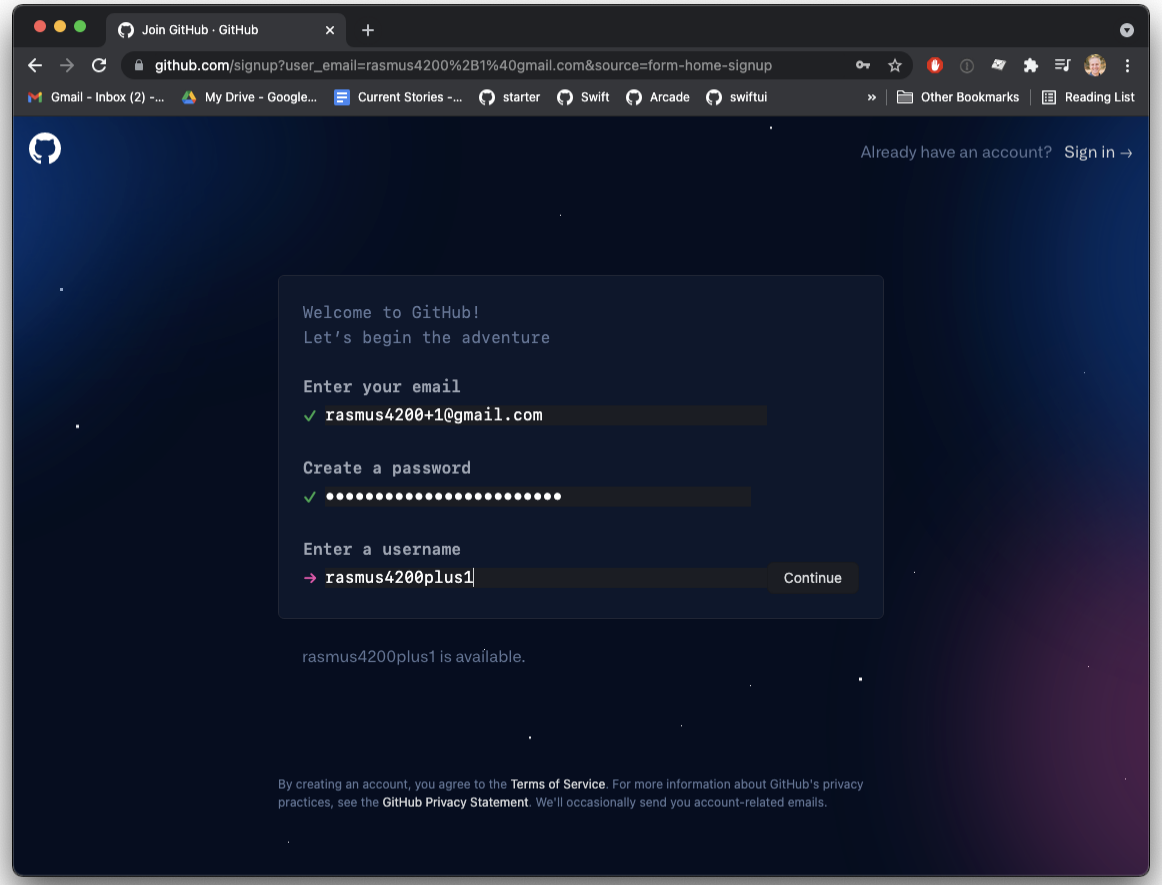
Git is the version control system most teams and companies use to store and collaborate on source code.

This document will walk you through how to set up your own git repos, so you can practice and learn the necessary git commands you will need to work professionally in the workplace.

Create your account

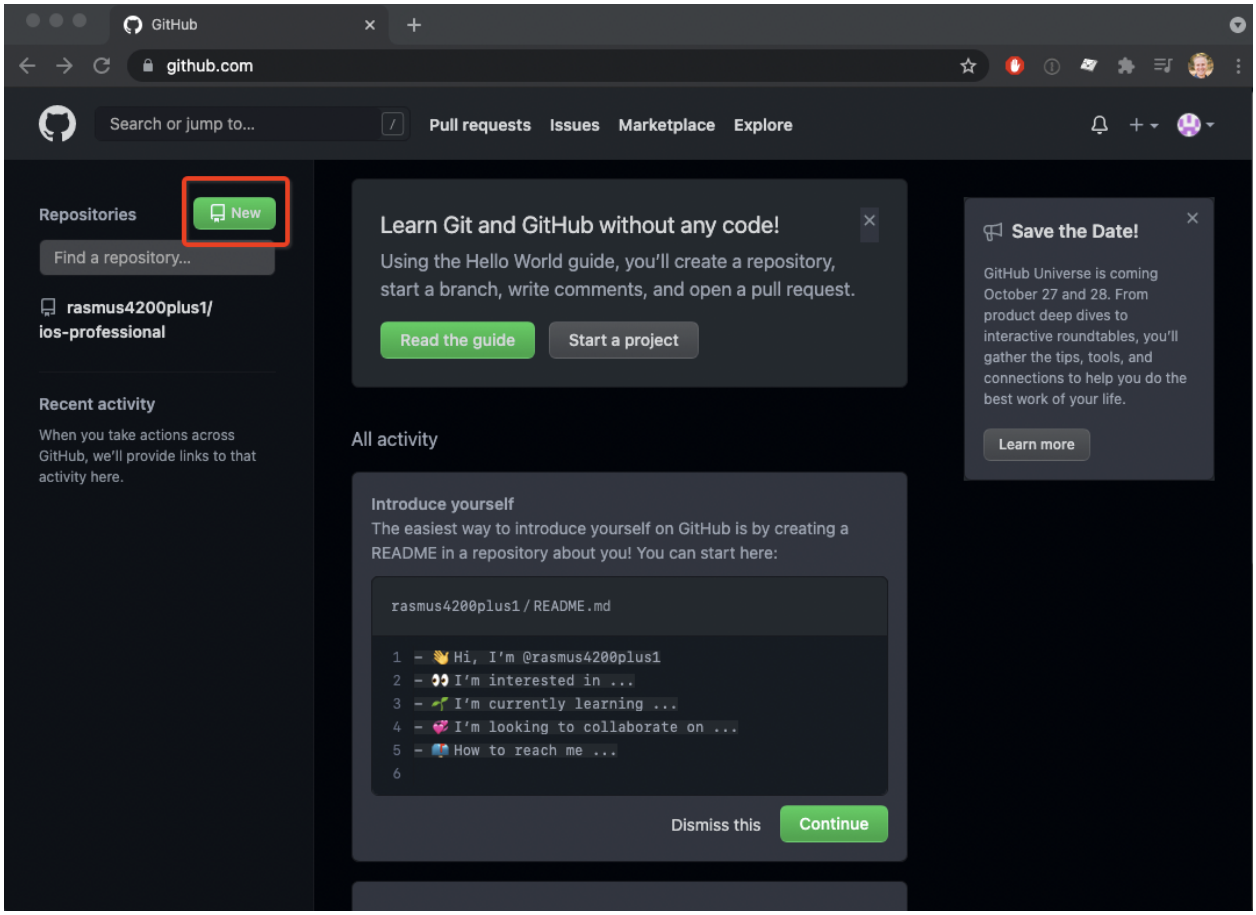
https://github.com/ signup

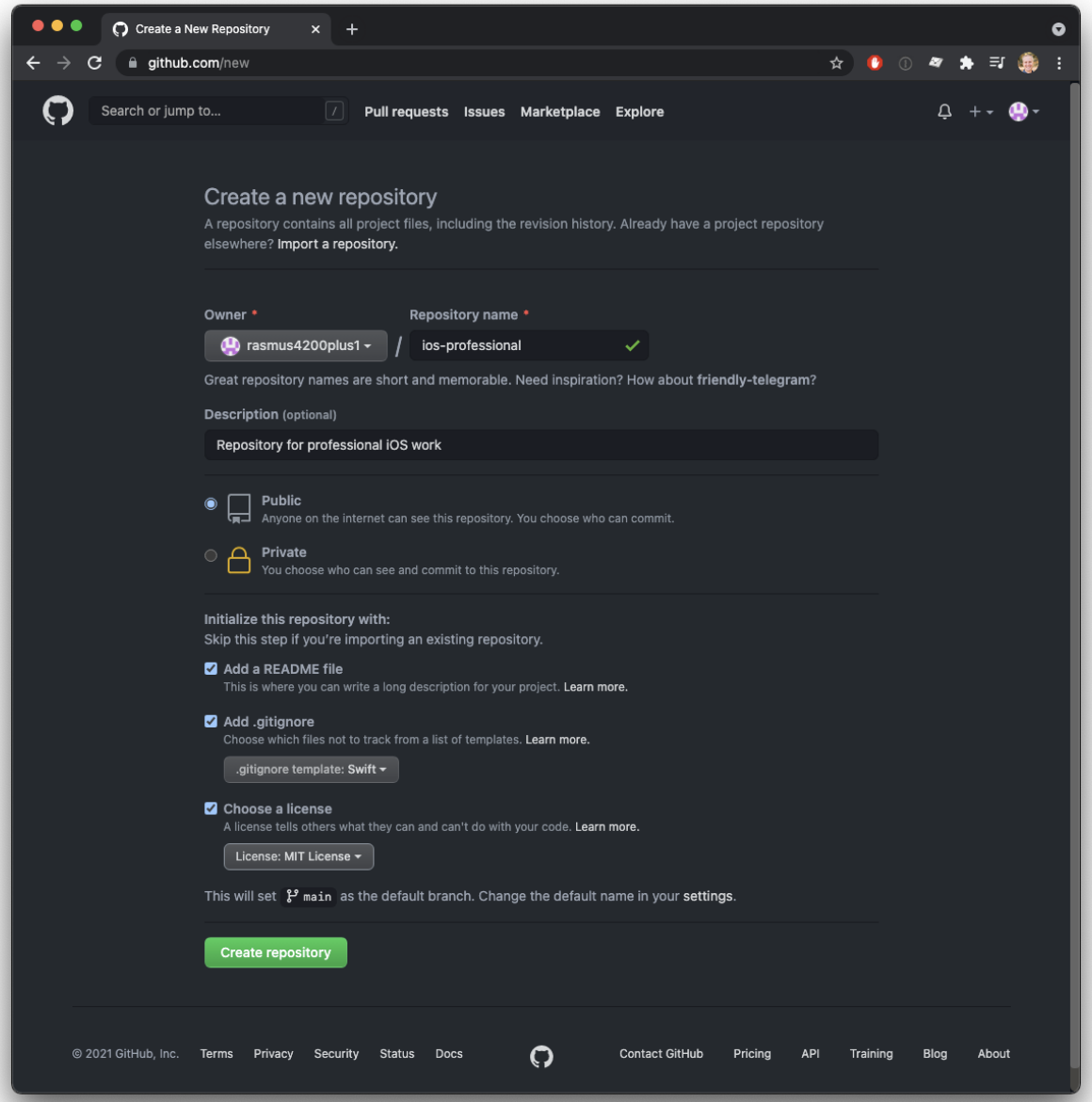


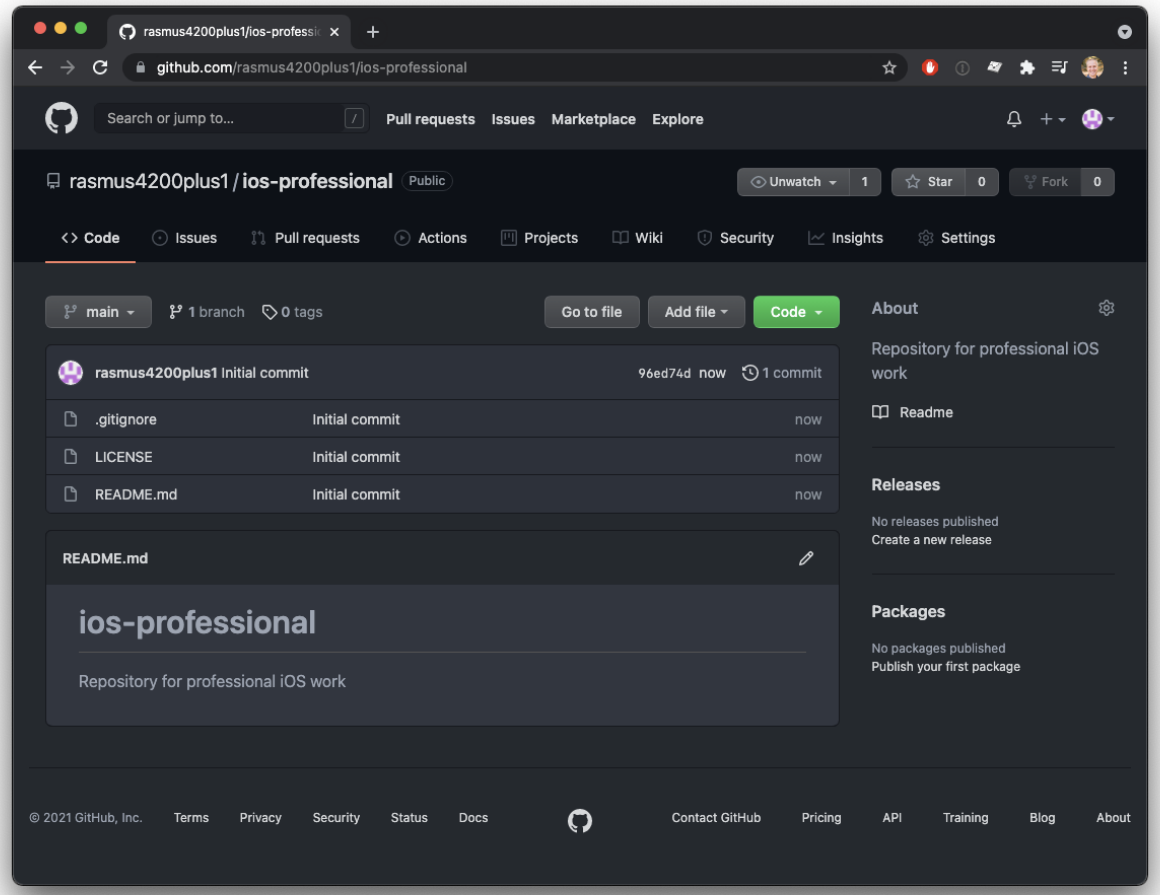

Join GitHub · GitHub
github.com/signup?user_email=rasmus4200%2B1%40gmail.com&source=form-home-signup
Gmail - Inbox (2) -...
My Drive - Google...
Current Stories -...
starter
Swift
Arcade
swiftui
Other Bookmarks
Reading List
Already have an account? Sign in →
Welcome to GitHub!
Let's begin the adventure
Enter your email
rasmus4200+1@gmail.com
Create a password
Enter a username
rasmus4200plus1
Continue
rasmus4200plus1 is available.
By creating an account, you agree to the Terms of Service. For more information about GitHub's privacy practices, see the GitHub Privacy Statement. We'll occasionally send you account-related emails.

Create your repository

Next let’s create your first repository.

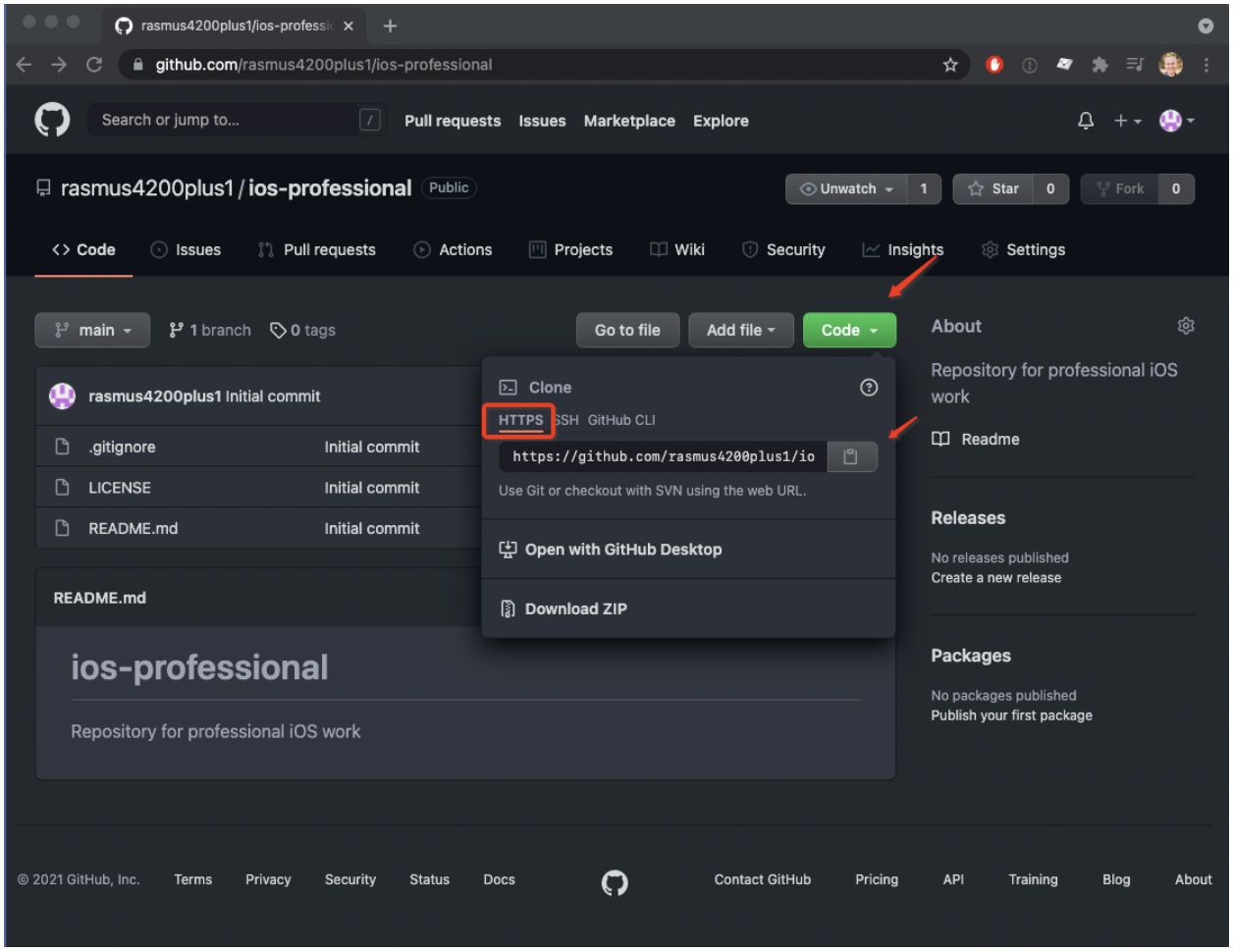



Create a New Repository
github.com/new
Search or jump to...
Pull requests
Issues
Marketplace
Explore
Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.
Owner *
Repository name *
rasmus4200plus1
ios-professional
Great repository names are short and memorable. Need inspiration? How about friendly-telegram?
Description (optional)
Repository for professional iOS work
Public
Anyone on the internet can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with:
Skip this step if you're importing an existing repository.
Add a README file
This is where you can write a long description for your project. Learn more.
Add .gitignore
Choose which files not to track from a list of templates. Learn more.
.gitignore template: Swift
Choose a license
A license tells others what they can and can't do with your code. Learn more.
License: MIT License
This will set main as the default branch. Change the default name in your settings.
Create repository
© 2021 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About


rasmus4200plus1/ios-professi
github.com/rasmus4200plus1/ios-professional
Search or jump to...
Pull requests
Issues
Marketplace
Explore
rasmus4200plus1 / ios-professional
Public
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
rasmus4200plus1 Initial commit
96ed74d now
1 commit
.gitignore Initial commit now
LICENSE Initial commit now
README.md Initial commit now
README.md
ios-professional
Repository for professional iOS work
About
Repository for professional iOS work
Readme
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
© 2021 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About

Checkout your repository

Now that we've created your repository, let's check it out locally to your machine.

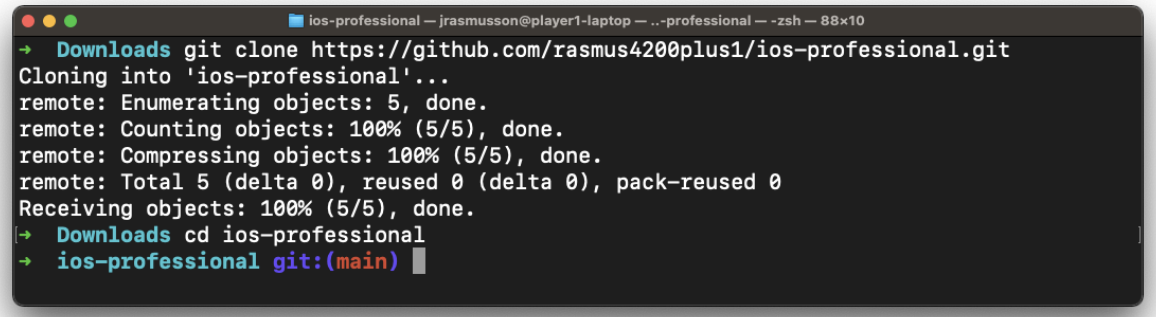


Click Code, HTTPS, and then copy the path to your repos.

Then open a terminal on your mac. Navigate to the directory where you want to checkout your repos, and clone your respos with the following command.

> git clone <repos path>
> cd ios-professional

And if all goes well you should see your repos on the branch called **main**.

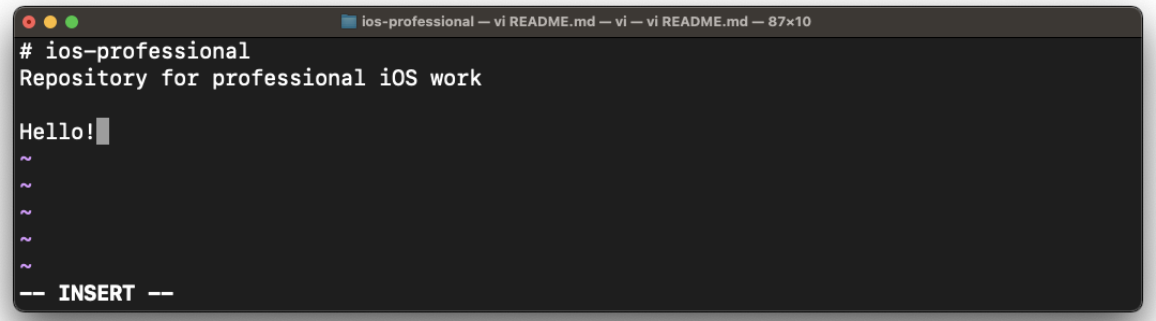

ios-professional — jrasmusson@player1-laptop — ..-professional — -zsh — 88×10
→ Downloads git clone https://github.com/rasmus4200plus1/ios-professional.git
Cloning into 'ios-professional'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
→ Downloads cd ios-professional
→ ios-professional git:(main)

Your first commit

To make a commit against your repository we first need to:

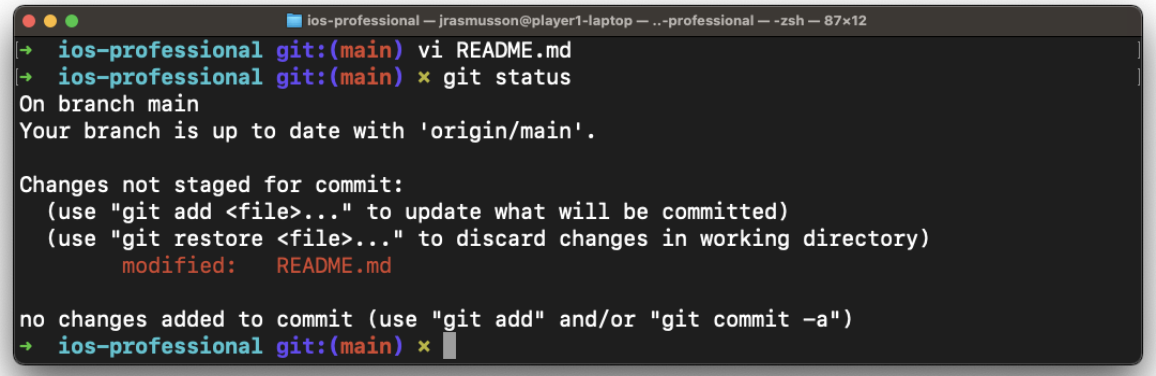
- Change a file
- Add it to our branch
- Commit it
- And then push it up to github

To change a file, open up the README.md file and make a change (i.e. add some text).



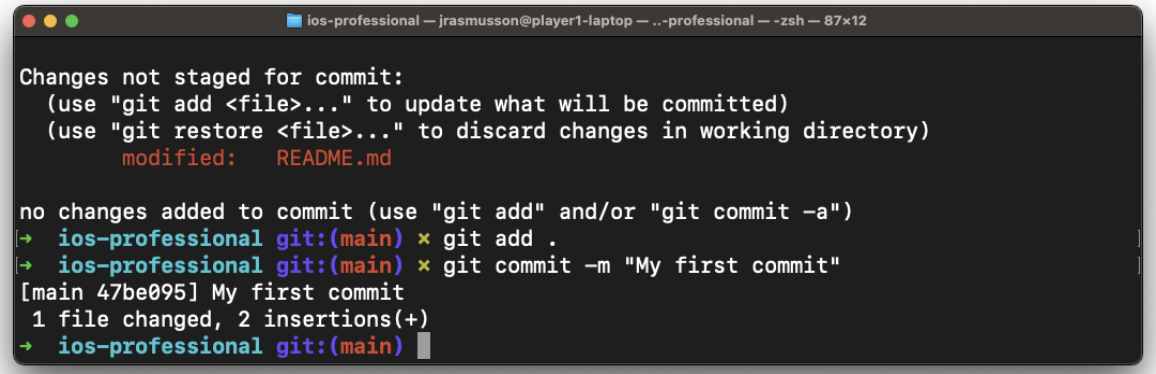
You can now see that a file has been changed on your repos by typing

> git status



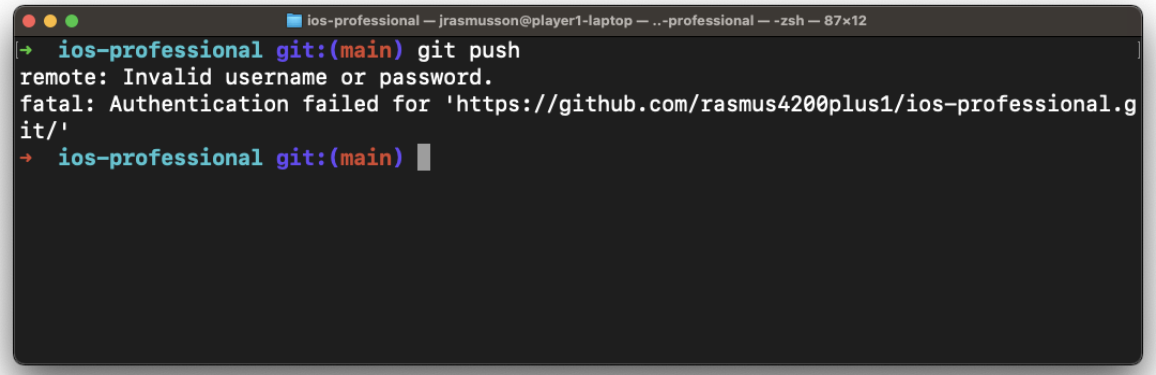
To add and commit that file type

> git add .
> git commit -m "My first commit"



With our changes committed, we are not ready to push them up to github. Try it with the following command.

> git push



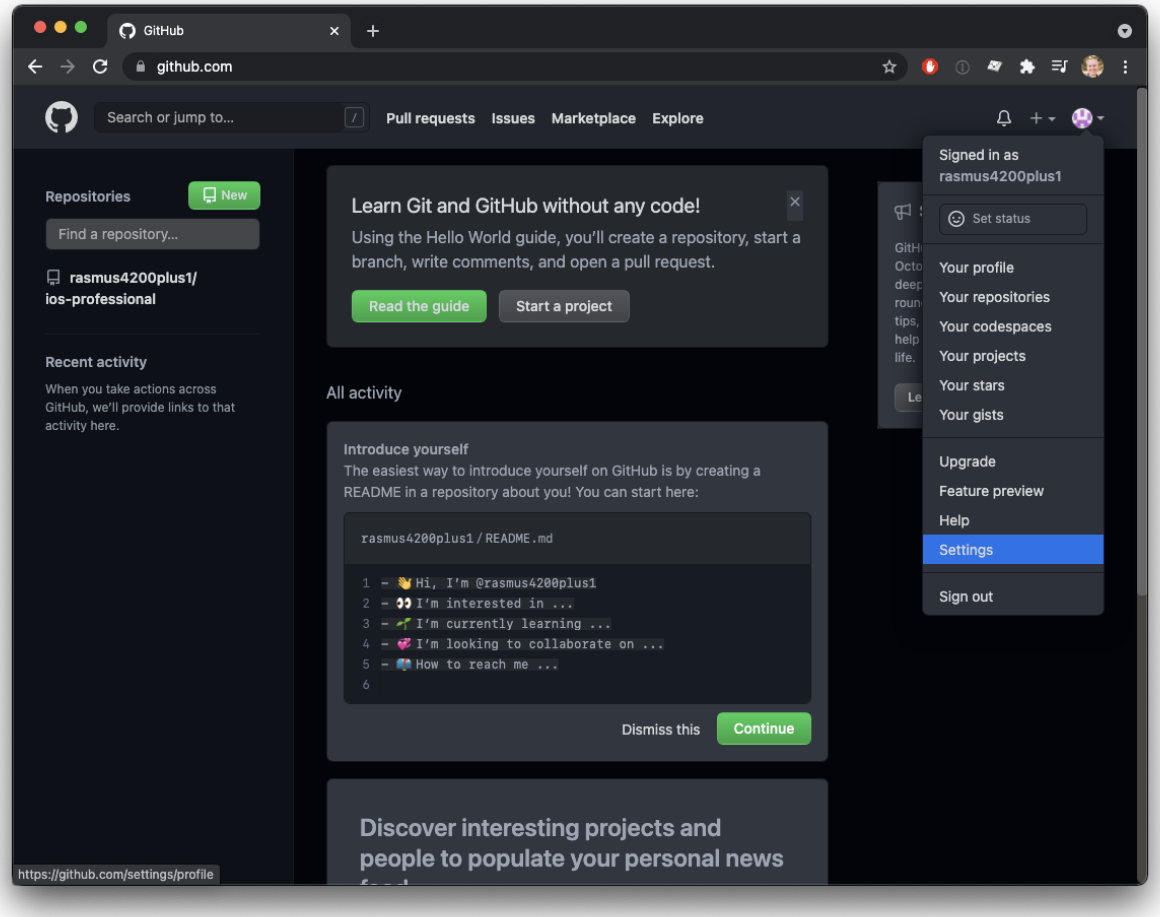
Normally this would work, but because it's our first time, github doesn't know how we are. So before we can do pushes, we first need to get a token.

Creating a personal access token

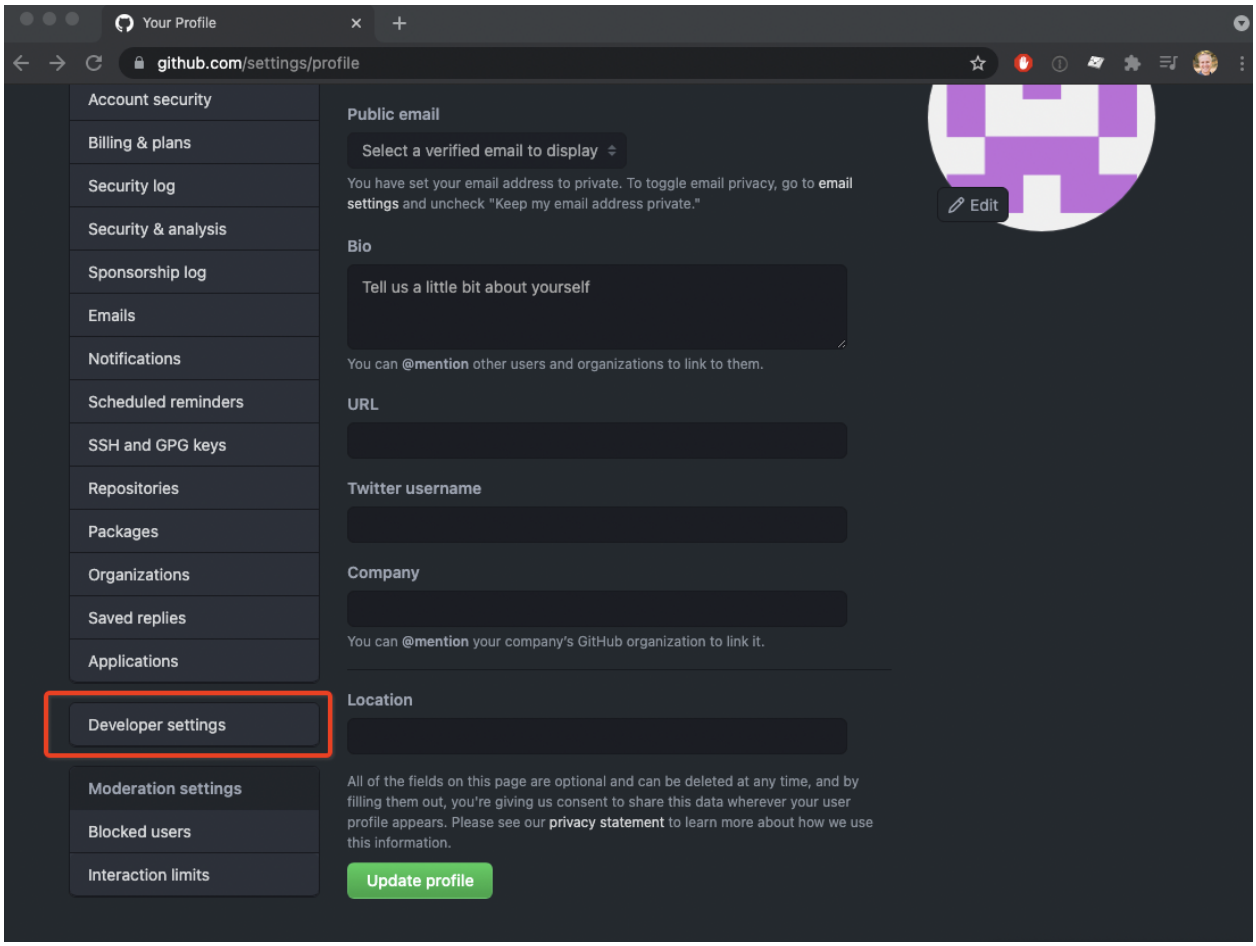
The steps for creating a token are outlined here. Let’s walk through them together.

First, if you haven’t already done so, open up your email and verify your email address.

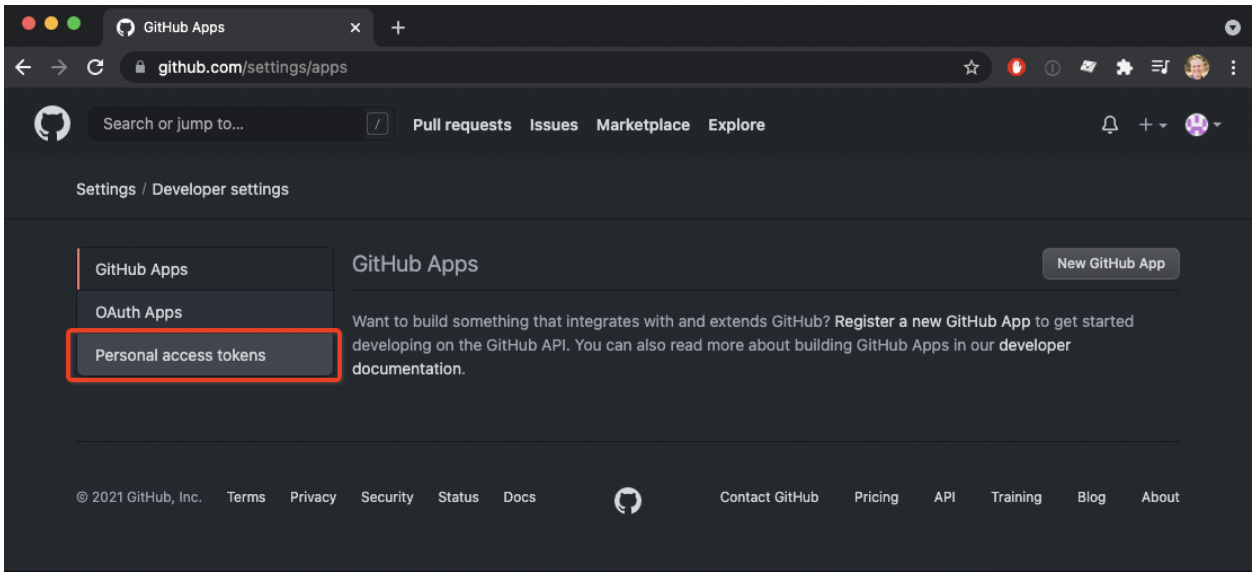
Then from the main github page, click Settings.

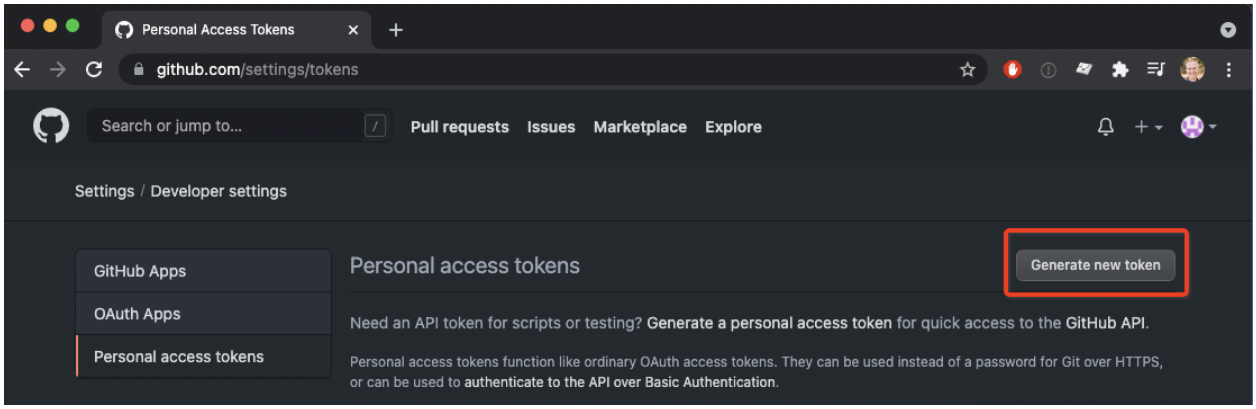


Click Developer settings



Person Access token

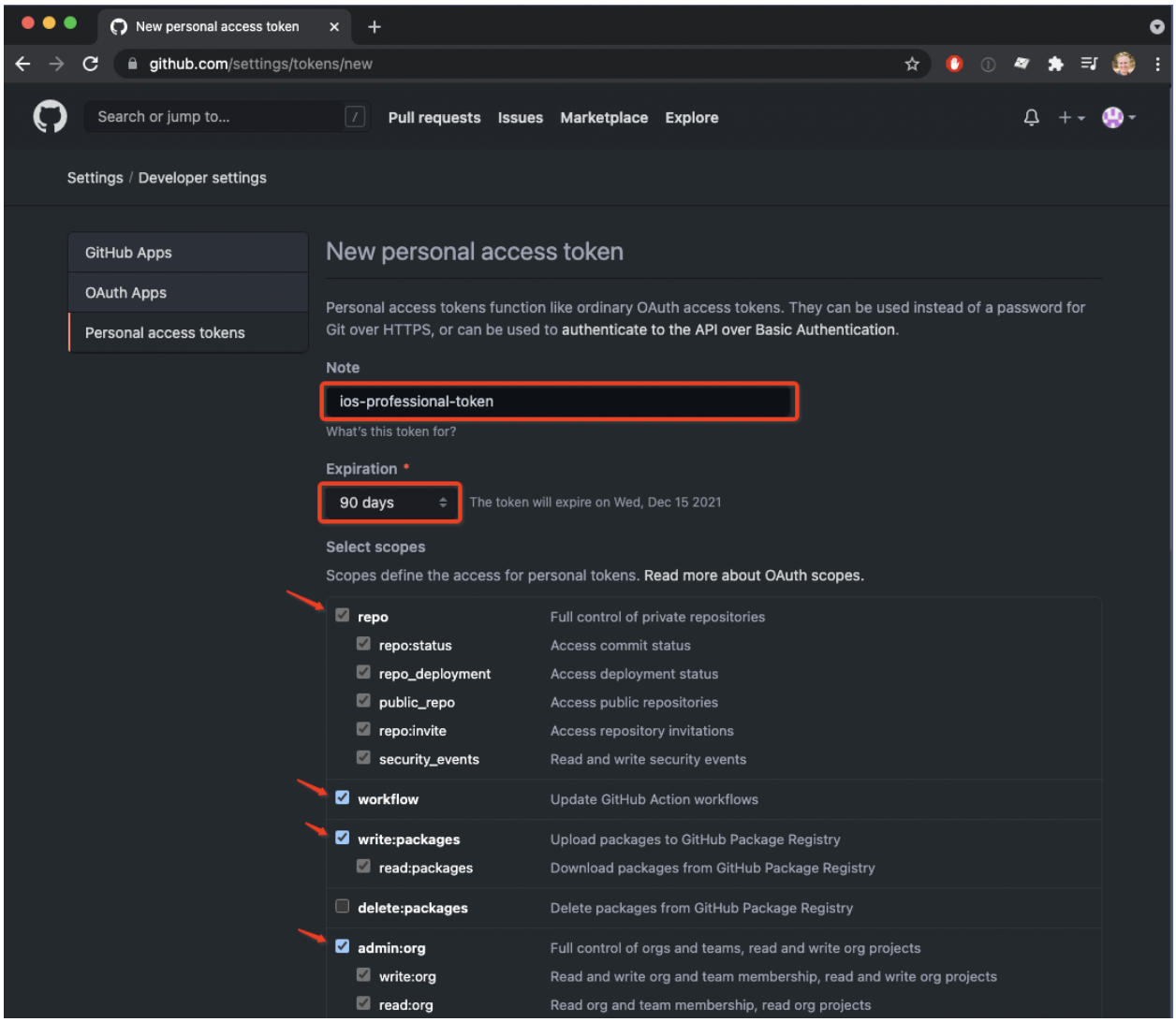




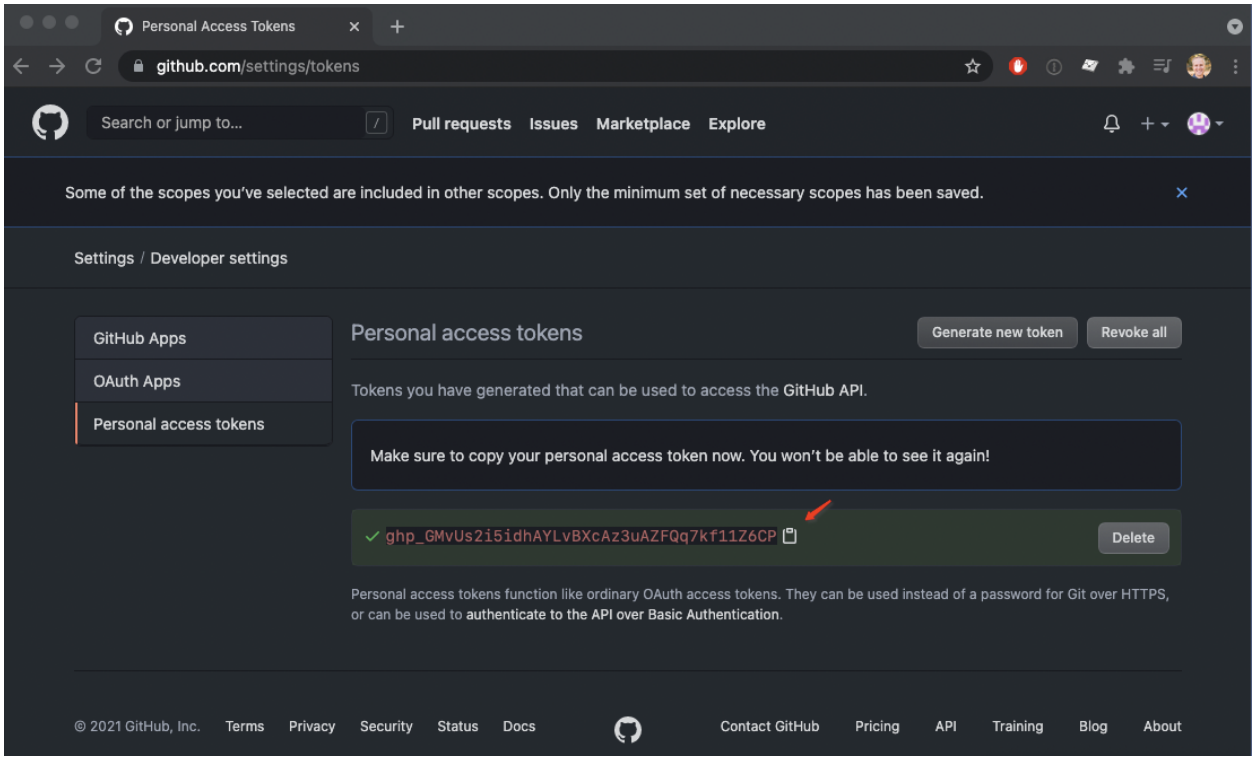
The official documentation for these steps can be found here.

Now we are ready to create our token. Give your token a name, expiration date, and necessary permissions to commit against your repository.

> ios-professional-token
> 90 days
> repos, workflow, write:packages, admin:org



Then scroll down to the bottom and click Generate token.



This token is what we are going to use to authenticate against github.

Copy it by clicking the copy button to the right of the code. Save it somewhere (or leave this window open).

And then to use it, we are going to install one more thing. The github command line tool or GitHubCLI.

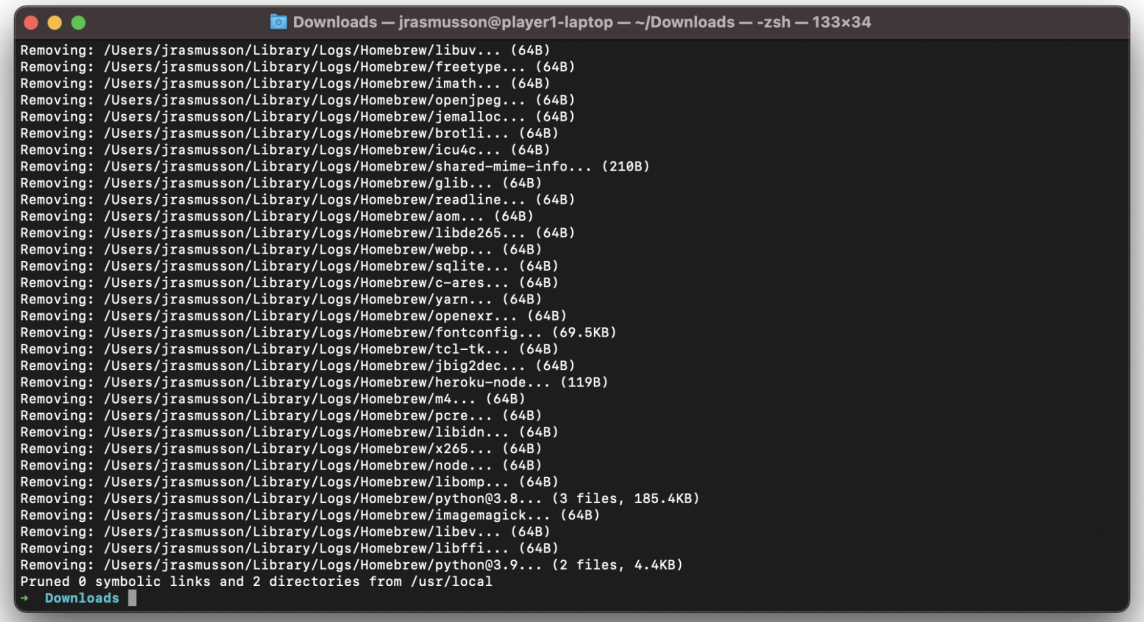
Installing GitHubCLI

In order to not have to enter our credentials every time, we need to cache our credentials somewhere. For this we are going to use GitHubCLI.

Instructions on installation options can be found here.

But for us, we are simply going to use homebrew (a mac installation tool). Open up a new command line terminal and type:

> brew install gh



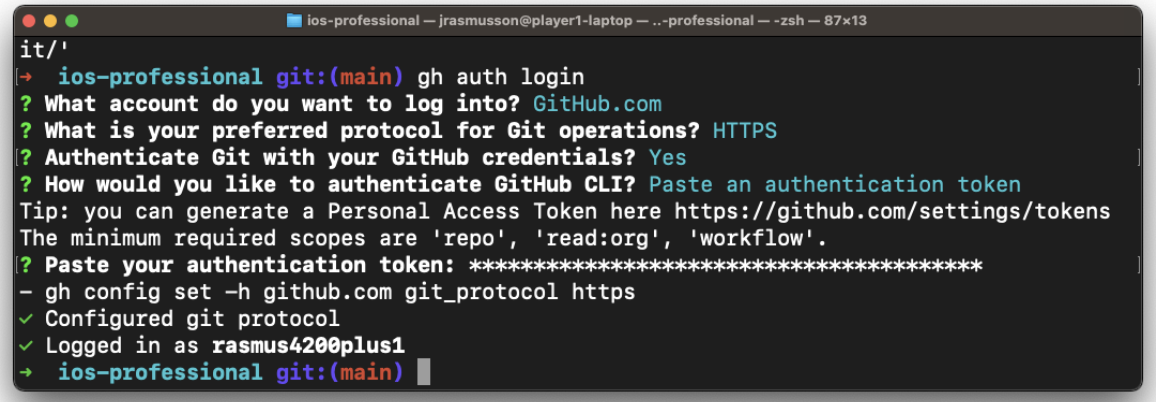
Enter GitHub Token

Then once GitHubCLI installation completes, enter your github credentials by typing:

> gh auth login

Follow the command prompts and use the following entries by pressing enter for each:

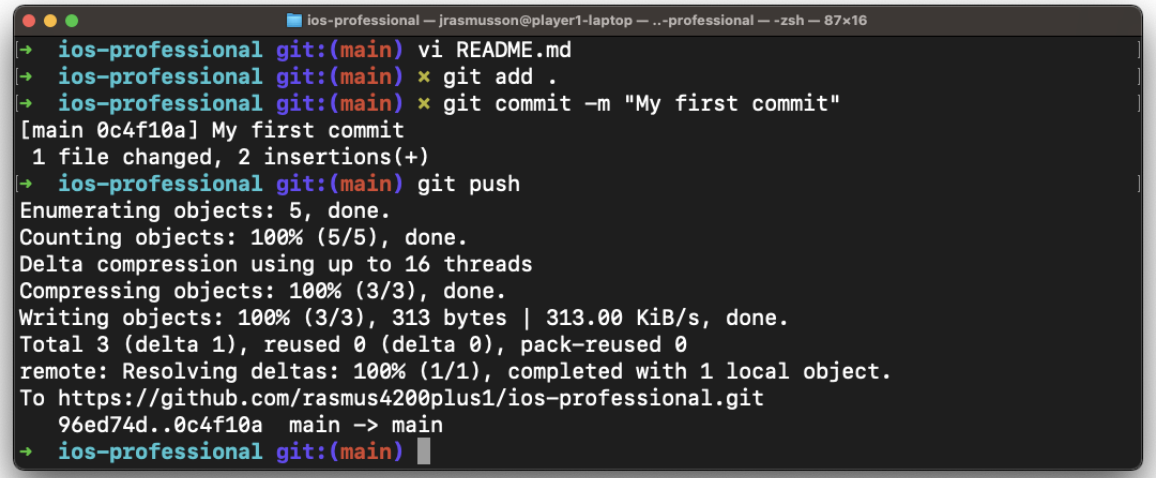
> GitHub.com (Enter)
> HTTPS
> Authenticate with credentials? Y
> Paste an authentication token



You should now be logged into github!

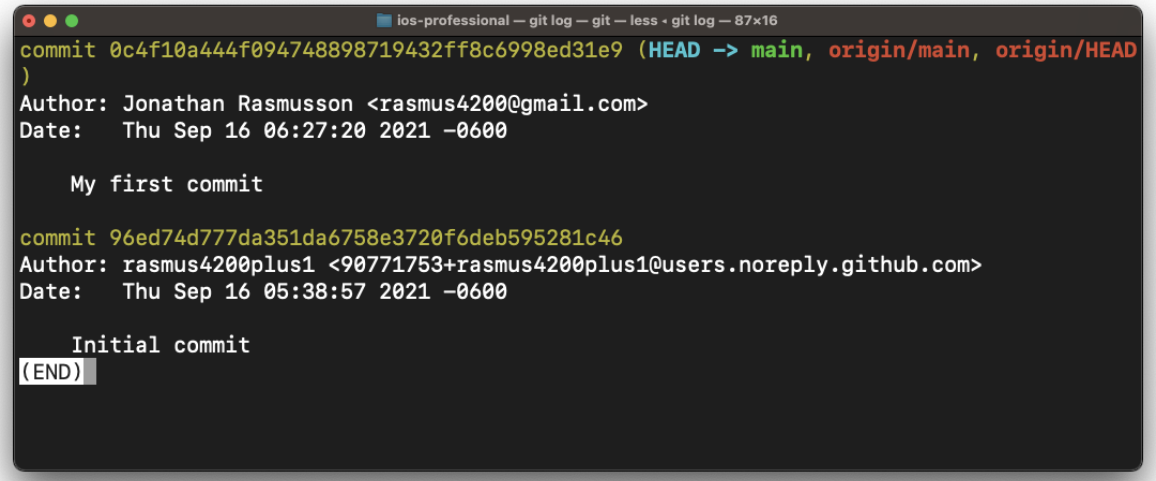
Now if we try making that push again.

> git push



It should commit. And if you type:

> git log



You should see your first commit stored and pushed against your repository.

Congrats!

What should I do when my token expires?

When you token expires, github will send you an email.

> We noticed your personal access token "ios-professional-token" with admin:org, repo, workflow, and write:packages scopes will expire in 6 days.
>
> If this token is still needed, visit https://github.com/settings/tokens/698099153/regenerate to generate an equivalent.
>
> If you run into problems, please contact support by visiting https://github.com/contact?tags=dotcom-accounts
>
> Thanks,
> The GitHub Team

Simply click the link. Generate a new token. And then follow the instructions above in the previous section `Enter GitHub Token`.