If you don’t already have a gihub account, go to the [signup page](https://github.com/join?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home) and create one.

Code has nice terminal feature for doing windows or mac commands. It is convenient to create a repository for your project using the command line feature. Click on

Inside code explorer open your project folder. My suggestion is that user create a separate folder for each project. My <projects home> is c:\users\gener\javascript\projects\ and this project is named simple-website and my user name is gener.

cd c:\<projects home>/simple-website

git init .

git config --global user.name <yourGitHubAccount>

git config --global user.email <yourGitHubEmailAccount>

Example:

git config --global user.name genericWorker

git config --global user.email genericWorker@gmail.com

git add .

git status

# edit .gitignore to ignore any folders you don't want // This is optional

git commit -m "first commit" // Commits whole folder

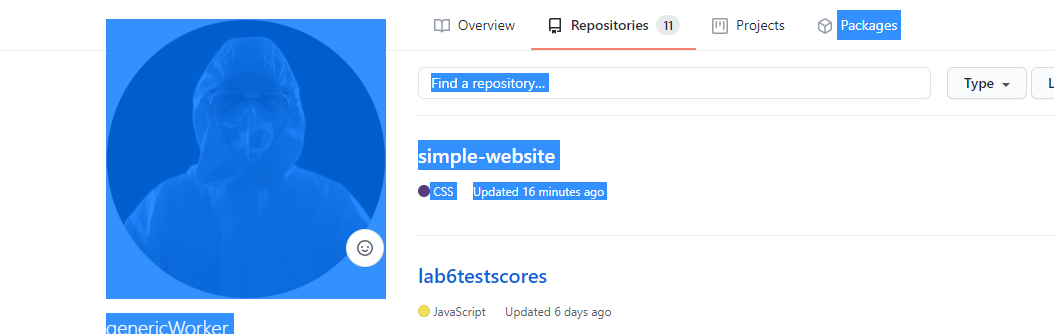
git remote add origin https://github.com/genericWorker/simple-website

git push -u origin main

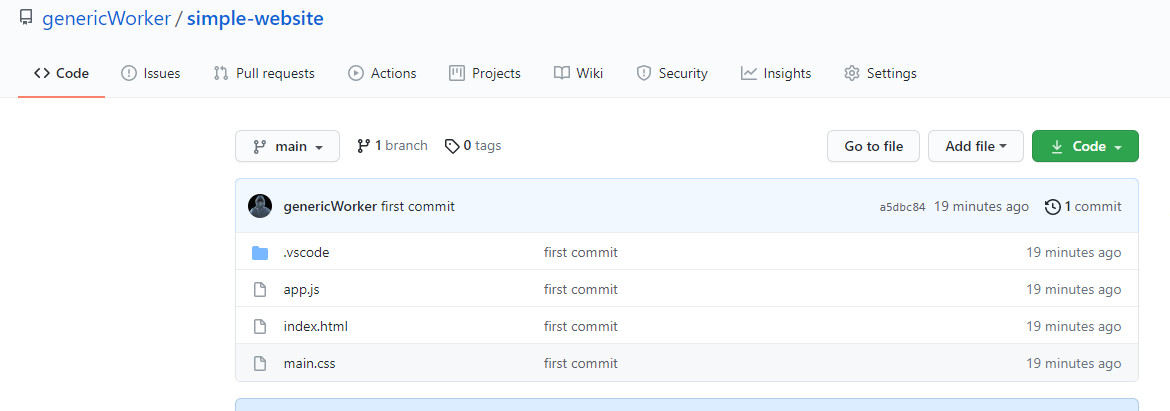
Open browser to check to that everything has been committed to you new repository:

<https://github.com/genericWorker>

Click on Repositories and look for your new repository “simple-website”. Click your repository link.



Click on the simple-website link to see the contents of your repository:



Congratulations on first javascript program and creating a repository for your app in github.

The next step is to deploy your app as a public website using the gitub platform.

Step 1:

Make a scratch directory somewhere in file system.

git clone <https://github.com/genericWorker/simple-website.git>

This will make a copy of your repository

Step 2:

git checkout –-orphan gh-pages

Step 3:

Not here ----git push -u origin gh-pages

git add .

git commit -m “update my site repo”git

git push -u origin gh-pages

Post local to new repo

cd c:\ABC

git igit config --global user.email <yourGitHubEmailAccount>

git add .

git status

# edit .gitignore to ignore folder you don't want

git commit -m "first commit"

cd ..

git push -u origin master

Project github pages

1. Start with a repository. Create a branch off that repository called gh-pages.
2. When branching use git branch – orphan gh-pages
3. Allowed one per project

Two ways to do it.

1. Automatic page generator
2. Bring your own code

Steps:

1. Create new repository for your project
2. Green New Button
3. Create special website genericworker.github.io (One of these per account)
4. Make the repository public (Click this)
5. Optional: create README.md
6. Click Settings Gear Icon and select Pages
7. This will show where site will be published for this repo (This is the account repo)
8. Optional – Use automatic page generator under New User Site
9. Optional – Continue to layouts to choose themes.

Github pages for Projects

1. Create my own repository, e.g. curlyMore, you can still can use README default
2. Copy url of my repository
3. Go to terminal and navigate to home directory for local repo. If you have a repository clone it to local directory.

git clone <https://genericWorker/simple-website.git>

cd simple-website

git checkout –orphan gh-pages

modify content if needed.

git add .

git commit -m “My initial commit”

// upstream remote back up to clone url

git push -u origin gh-pages

// Now in repo, we will be able to see gh-pages branch in the repo

Make any modifications then

git status // check of anychanges

git add .

git commit -m “message”

git push // To push changes up to repo (still part of gh-pages branch)

// page should now be live on (Wait up to 10 minutes)

<https://genericworker.github.io/simple-website/>

Other sites: on training.github.com/classes:

Introduction to github

GitHub for Everyone