

[Get started](#)[Open in app](#)

## Kishandth SIVAPALASUNDARAM

[Follow](#)

4 Followers

[About](#)

## How to deploy your React-App to a Github User Page



Kishandth SIVAPALASUNDARAM Feb 16, 2020 · 4 min read

The other week I was attempting to deploy my React-App portfolio website to a Github User Page (not Project Page) and found that it is not straight-forward so I've written a detailed guide explaining how to do it.

### Difference between Project Pages and User Pages on Github

Github has two types of Github pages you can deploy onto. The first is a project page and the other is a user page.

A user page is one which is a subdomain of Github that looks like this:

`<github username>.github.io`

[Get started](#)[Open in app](#)

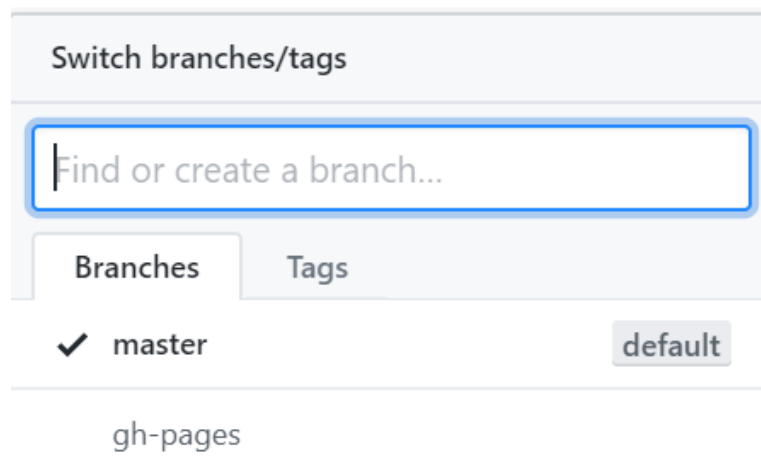
<github username>.github.io/<project name>

## Deploying to a Project Page on Github

Deploying a React App to a project page on Github is very straight-forward. If you are using Create-React-App you can simply run `npm run build` then add `"deploy": "gh-pages -d build"` to the `scripts` part of your `package.json`. Also add `"homepage": "https://<github username>.github.io/<project name>"` to your `package.json` as well.

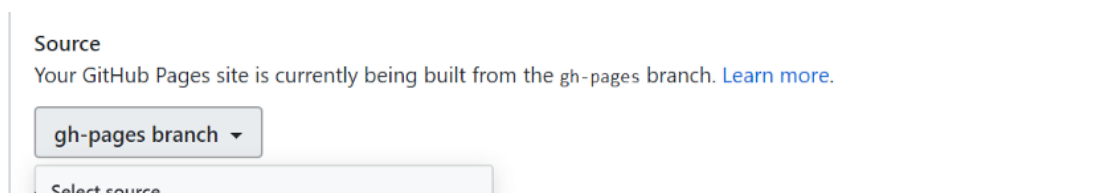
Install Github Pages using `npm install gh-pages --save-dev` after which you can run `npm run deploy` or `yarn deploy` to send your project to Github.

This will create a new branch in your Github repository that looks a little like this:



The master branch contains your original code and the gh-pages branch contains your deployed code.

The final step is to tell Github to use the deployed version of the repository in the `gh-pages` branch instead of the master branch when users navigate to your Github page. To do this go to the 'Settings' tab of your repository in Github and navigate down till you see the Github Pages section. Choose the `gh-pages` branch instead of the `master` branch.



[Get started](#)[Open in app](#)

|  |   |
|--|---|
| <b>master branch</b><br>Use the master branch for GitHub Pages.                  |   |
| <b>master branch /docs folder</b><br>Use only the /docs folder for GitHub Pages. | from a domain other than kish.work. <a href="#">Learn more.</a> |
| <b>None</b><br>Disable GitHub Pages.   |   |

☒ **Enforce HTTPS**  
HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

You can now view your new github pages site at: `https://<github`

`username>.github.io/<project name>`

You can find a more detailed explanation on how to deploy your project onto a Github Project Page [here](#).

## Challenges with deploying onto a Github User Page

Github user pages are made to only use the master branch as you can see from the screenshot below.

### Source

Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

User pages must be built from the master branch.

If you use a starter like `create-react-app` your `index.html` will be located within the `public` folder, which Github won't read. Github only reads `Readme.md` or `.html` files located in the upper-most layer of your app. This creates a very frustrating issue where you will have to relocate your `index.html` file onto the root and then change all of the references in your React App to point to the new location of `index.html`.

## An easy trick

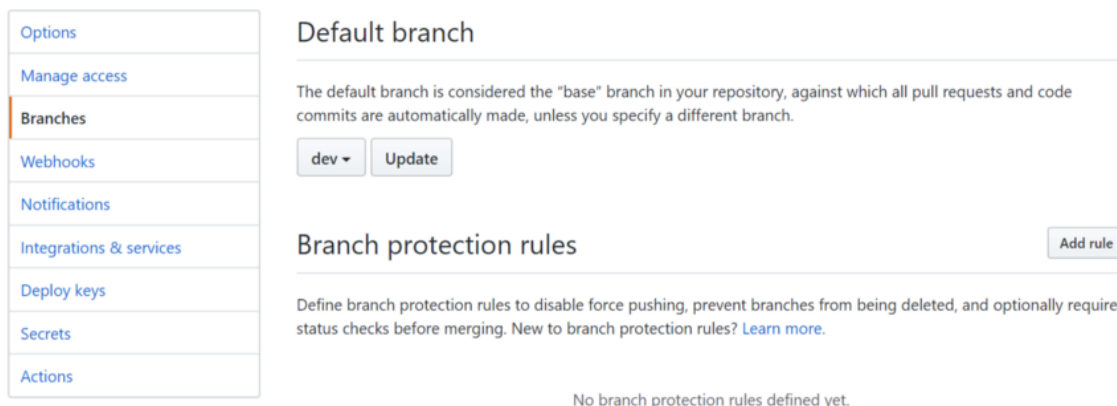
A very easy and quick solution to the problem above is to move your repository from the master branch to a new branch and then deploy your build to the master branch.

The code below will create a new branch and call it `dev`. The line below it will push the branch to Github along with the code saved locally.

```
git checkout -b dev
git push -u origin dev
```

[Get started](#)[Open in app](#)

Github. Now head to 'Settings' in your Github repository and find the 'Branches' tab on the Navigation bar to the left. Now set your default branch to 'dev' as seen in the screenshot below and click Update.



This will now make the `dev` branch the default branch for your Github repository and this will allow you to deploy your build to the `master` branch.

The final step is to add `"deploy": "gh-pages -b master -d build"` to the `"scripts"` section of the `package.json`. Then when you run `yarn deploy` the build version will be deployed to your master file. Well done! Your React App should now be visible on `<github username>.github.io`.

[React](#) [Create React App](#) [NPM](#) [Github](#) [Git](#)

[About](#) [Help](#) [Legal](#)

Get the Medium app

