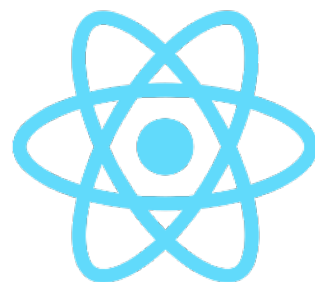


# deploying react.js

---

{codenation}<sup>®</sup>



# intro: {

**It can be confusing when you have finished your react app and it is time to put it out there on the web because its not as simple as before (especially GitHub-pages).**

**Hopefully this brief, succinct guide will help you!**

}



# GitHub-Pages: {

This is very straightforward process! However it is different to just publishing a normal portfolio.

If you have used GitHub Pages before then you will have followed these steps:

Repository > Settings > GitHub Pages > Source > Master Branch

## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <https://benmaudslay.github.io/cn-template-portfolio/>

### Source

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

master branch ▾

## The react way:

**First things first:** Add, commit and push your latest code to the repository.

Now that we've got that out the way, let's begin:

### 1. Add homepage

Open your package.json file inside your react app and add the **homepage** property.

If you look closely you'll see that the package.json file only contains an object, so therefore we can add properties (key-value pairs).

Now in the **homepage** property you need to add the url that the site will be found on. You will need to get the url of your GitHub repo as if we are using the old of publishing a repo, it will look like this:

**`https://username.github.io/repo-name/`**

The end result will look like this:

```
"homepage": "https://yourusername.github.io/repository-name"
```

## 2. gh-pages

Next, we need to install a package called gh-pages.

Run this in your terminal:

```
npm install --save gh-pages
```

## 3. Add some scripts

If you take a look you will see a property called `scripts` inside your `package.json` file. It looks like this:

```
"scripts": {  
  "start": "react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject"  
}
```

Now we need to add these to it:

```
"predeploy": "npm run build",  
"deploy": "gh-pages -d build",
```

It should look like this when you are done:

```
"scripts": {  
  "start": "react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject",  
  "predeploy": "npm run build",  
  "deploy": "gh-pages -d build"  
}
```

## 4. Run Deploy

Now run `npm run deploy`

## 5. Setup source to gh-pages branch

Once you have successfully deployed. A build folder will appear in your file system.

Open your GitHub repository and click on settings tab, if you scroll down you can see the GitHub Pages section. Then change the branch to gh-pages.

### GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <https://saigowthamr.github.io/React-Accordion/>

#### Source

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

gh-pages branch ▾

Save

#### Theme Chooser

Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

Custom domain

