



How To Deploy React Application On GitHub Pages



Ganesan C



Updated date Sep 09, 2021



5.1k



2



9



[Download Free .NET & JAVA Office Files API](#)
[Try Free File Format APIs for Word/Excel/PDF](#)

Introduction

In this article, we are going to explore how to deploy the React application on GitHub pages.

What are GitHub pages?

- GitHub Pages is a **static website hosting service** and it is **free** of charge.
- It takes resources like HTML, CSS, and JavaScript files directly from the **GitHub repository**.
- We can host our website on GitHub's with **github.io** domain or our own custom domain name.

Types of GitHub Pages

- There are three types of GitHub Pages sites which are,
 - User site
 - Project site
 - Organization site.
- Project sites are connected to a specific project hosted on GitHub, such as a JavaScript library.
- User and organization sites are connected to a specific GitHub account.

User Site

- The user site is connected to a specific GitHub account.
- You should have a user account on GitHub.
- To publish a user site, you must create a GitHub repo owned by your user account and repo name should be **username.github.io**. For Example, If your user name is **TestName** your repo name should be **testname.github.io**.
- If you are not using any custom domain, then your website will be available at **https://username.github.io**.

Organization Site

- The organization site is connected to a specific GitHub account.
- To publish an organization site, you must create a repo owned by an organization and the repo name should be **organization.github.io**.
- If you are not using any custom domain, then your website will be available at **https://organization.github.io**.
- You can create only one user or organization site for each user account on GitHub. But we can create unlimited Project sites which can be owned by an organization or a user account.

How to Publish the React application on GitHub Pages



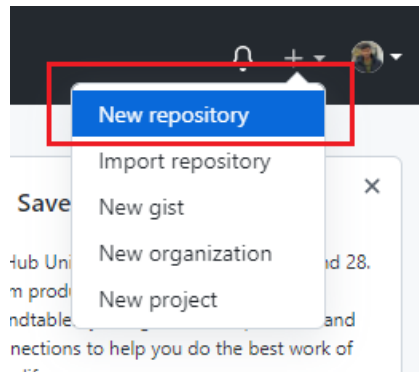
1. Create an Account in GitHub.
2. Create Repo in GitHub where the repo name should be **username.github.io**
3. Create React Application.
4. Deploy the React Application using GitHub Pages.
5. Commit and Push the codebase (React Application) into GitHub repo.

Step 1 - Create an Account in GitHub

- Open the GitHub link
- Create an account using your personal email
- To secure your GitHub account, you can enable MFA (multi-factor authentication) login for your account.

Step 2 - Create Repo in GitHub

- Login to your GitHub account.
- Create the new repo by clicking the "+" icon at the right top of the page.




- Enter the repo name as **yourusername.github.io** and choose whether the repo should be public or private.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner *

 Ganesan-Chandran ▾

Repository name *

Great repository names are short and memorable. Need inspiration? How about [fluffy-tribble?](#)

Description (optional)

☒



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.](#)

☐

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

- Click on the "Create repository" button.



Step 3 - Create React Application

Create the react application which you want to host as a website. In this article, I am mainly focusing on how to deploy the react application on GitHub Pages. So please refer to this [link](#) to create React application.

Step 4 - Deploy the React Application

To deploy the application, follow the below steps.

1) Add GitHub Pages dependency package

Install "gh-pages" package using the below command.

```
1 npm install gh-pages --save-dev double --save-dev
```

2) Add homepage property to package.json file

- Add the below property to your package.json file.
- For a GitHub user site:
 - **"homepage": "https://{username}.github.io"**
- For a custom domain:
 - **"homepage": "https://testwebsite.com"**

```
{
  "name": "ganesan-chandran-profile",
  "version": "1.0.0",
  "private": true,
  "homepage": "https://ganesan-chandran.github.io",
  "dependencies": {
    "gh-pages": "^3.2.3",
    "react": "^17.0.2",
    "react-dom": "^17.0.2",
    "react-scripts": "4.0.3",
    "react-scroll": "^1.8.3"
  },
  "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"
  }
}
```

3) Add deploy scripts to package.json file

Add both predeploy and deploy property scripts to the package.json file as below,

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

The "predeploy" command is used to bundle the react application and the "deploy" command helps to deploy the bundled file.

```
"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) Create a remote GitHub repository

- Initialize the Git using **"git init"** command.
- Add it as remote using **"git remote add origin your-github-repository-url.git"** command.

5) Deploy the Application to GitHub Pages

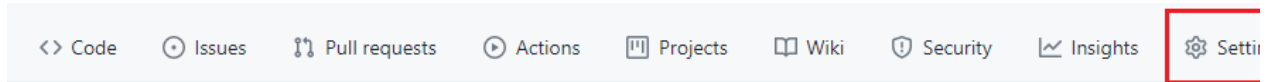


```
1 npm run deploy
```

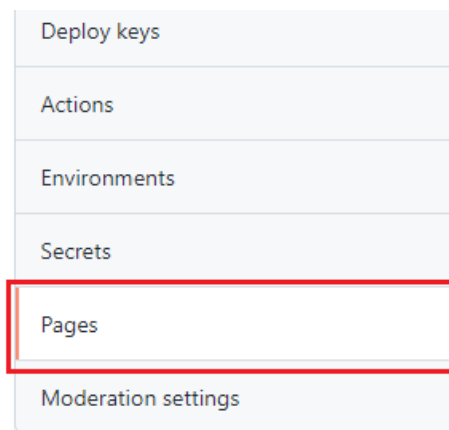
6) Access deployed site

To get the published URL,

- Go to your GitHub Repo.
- Click Settings menu.



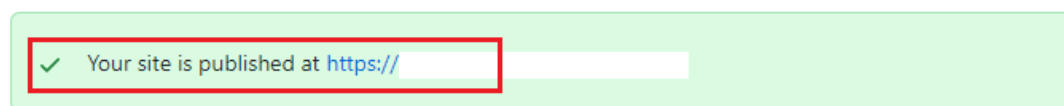
- Go to the "Pages" Section.



- You can see the **"Your site is published"** message.
- Select branch to **"gh-pages"** and click on the **"Save"** button.

GitHub Pages

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



Source

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



Now, you can access the deployed site using the published URL!

Step 5 - Commit and Push the codebase

Commit and push your code changes to the GitHub repo. It is an optional one. But, for maintaining the code base we can do this step.

```
1 git add .
2 git commit -m "commit message"
3 git push origin master
```

[deploy on github pages](#)[deploy react applications](#)[github pages](#)