

Photo by [eiitiaoki](#)

# GitHub Pages でReact Appを公開する方法



UI/UXデザイナー うっくん

2024年2月7日 12:44

## 背景

2019年ごろより、自身のPortfolio兼自己紹介用のWebページとして、GitHub Pagesという仕組みを使って無料でWebサイトをホスティングすることを行なっています。

### GitHub Pagesでデザインポートフォリオ作んのオススメやで！

はいまいど～！うっくんです！みなさんポートフォリオって持ってますか？それなんぞ？と言う人のために簡単に言うと、自分の作品集のようなものなのです。デザイナーの場合、就職や転職活動をするときに提出を求められるケース...

♡ 17



UI/UXデザイナー うっくん

2019/04/24 21:59



2022年ごろにはさらに一步踏みこんで、Reactで作ったSPAをGithub Pagesの仕組みを使ってホスティングするようにしていました。こうすることで↑で紹介しているような静的ページ（いわゆる「ペライ

チ」)ではなく、複数のページをリンクして本格的なアプリケーションを無料で公開できるというメリットがあります。他にもReactコミュニティのさまざまなライブラリを活用してMaterial UIなどのUIフレームワークを使ってリッチなUIを作り込んだり、i18nextというライブラリを使って多言語対応などを行なっています。

私は、Reactの勉強も兼ねてこれを [wecken.github.io/portfolio](https://wecken.github.io/portfolio) というURLで閲覧できるようにしていました。

本来は [wecken.github.io](https://wecken.github.io) の直下でホスティングしたいのですが、設定をミスって2024/02/07 現在は見れない状態になっていました。しょうがないので、ホスト先を /portfolio にすることでなんとか現在は確認できるようになっているということです。

さらに悪いことに、色々トラブルシューティングをやっている間に、結構あかん失敗をしてしまったみたいで、新規の更新ができない感じになっていました（これは以前同僚に教えてもらった時に達した結論なので、実際どの辺で何をやってジ・エンドになったのかは自分ではよくわかりません）

この度、これを最初からやり直して、wecken.gitub.io 直下で見えるようにしようと決意したので、同じ過ちを起こさないように手順をここに記録したいと思います。

## ▼ 目次

背景

---

やったこと

---

GitHub Pages でReactAppを公開するまでの手順

---

1. Reactアプリケーションの準備
  2. GitHubリポジトリの準備
  3. Reactアプリケーションのビルド
  4. GitHub Pagesへのデプロイ
  5. GitHubリポジトリの設定
- 

完成！

---

GitHub PagesでホスティングしているReact Appを更新する手順

すべて表示

# やったこと

## ChatGPT

A conversational AI system that listens, learns, and challeng  
chat.openai.com

上のようにChatGPT に聞きながらやります。

もしエラーとかに当たってもその場で聞けば教えてくれるし、隣に詳しいエンジニアを座らせて教えてもらっているのとかかなり近い体験になっています。

上のリンクを読んでもらうと分かりますが、大まかな流れとしては

1. LocalでCreate React Appをして、React Appのテンプレをインストールする
2. そのディレクトリをGitHubのリポジトリとして新規登録する
3. そのリポジトリでGitHub Pagesの設定を行う
4. ローカルで編集して、内容を更新したら都度、ビルド&デプロイのコマンドを打つ

という感じです。

多分以前の私は4のところでミスってしまい、デプロイコマンドを打たずにプッシュしてしまったとかそんな感じのことで事故ってしまった記憶がおぼろげにあります。

何ヶ月もポートフォリオを触っていないところにいきなり更新しようとしたので、手順を完全に失念していたのでしょう・・・。

# GitHub Pages でReactAppを公開するまでの手順

GitHub Pagesを使用して**username.github.io**直下でReactアプリケーションをデプロイするプロセスは、以下のステップに分けて実行できます。これにより、あなたのReactアプリケーションがGitHubのユーザー専用ページとして公開されます。

## 1. Reactアプリケーションの準備

まず、ローカル環境でReactアプリケーションを作成または準備します。もし新規にプロジェクトを作成する場合は、以下のコマンドを使用して新しいReactアプリを作成します。

```
npx create-react-app my-app cd my-app
```

TypeScript環境でやりたい場合は、テンプレートのオプションをつけます

```
npx create-react-app my-app --template typescript
```

このコマンドは**my-app**ディレクトリを作成し、Reactアプリの雛形を設定します。

## 2. GitHubリポジトリの準備

- GitHubにログインし、新しいリポジトリを作成します。リポジトリ名は**username.github.io**とします。ここでの**username**はあなたのGitHubユーザー名に置き換えてください。
- リポジトリが作成されたら、ローカルのReactアプリケーションディレクトリからGitHubリポジトリにプッシュします。

## 3. Reactアプリケーションのビルド

Reactアプリケーションをビルドするために、以下のコマンドを実行します。

```
npm run build
```

このコマンドは**build**ディレクトリを作成し、プロダクション用のアプリをビルドします。

## 4. GitHub Pagesへのデプロイ

- **gh-pages** パッケージのインストール: Reactアプリを簡単にデプロイするために、**gh-pages**というツールを使用します。以下のコマンドでインストールできます。

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

- **package.jsonの編集:** プロジェクトの**package.json**ファイルを開き、**homepage**フィールドを追加します。これはあなたのGitHub PagesのURLになります。

```
"homepage": "http://username.github.io",
```

- **デプロイスクリプトの追加:** さらに、**package.json**にデプロイ用のスクリプトを追加します。

```
"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"  
}
```

- **デプロイ:** 以下のコマンドを実行して、アプリケーションをGitHub Pagesにデプロイします。

```
npm run deploy
```

このコマンドは**build**ディレクトリの内容を**gh-pages**ブランチにプッシュし、GitHub Pagesがこのブランチを公開するように設定します。

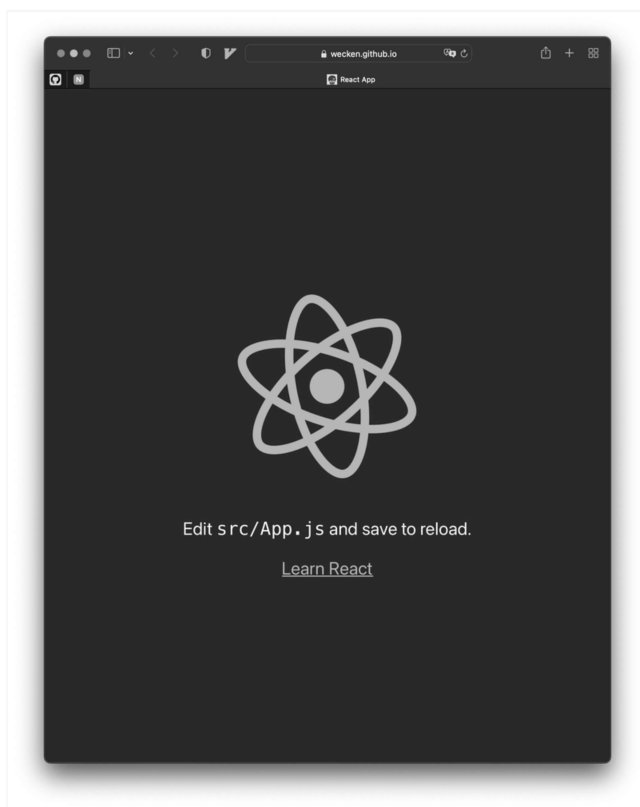
## 5. GitHubリポジトリの設定

- GitHubのリポジトリ設定にアクセスし、「Pages」セクションを見つけます。
- 「Source」セクションで、**gh-pages**ブランチを公開ソースとして選択し、保存します。

これらのステップを完了すると、あなたのReactアプリケーションは**username.github.io**でアクセス可能になります。変更が反映されるまでには数分かかる場合があります。

## 完成！

と、この記事を書いている間にとりあえずReactAppをGitHub Pagesでホスティングすることに成功しました。



ここまでは非常に簡単ですね。ある程度GitとかGitHubに慣れている人なら10分ぐらいでできると思います。初心者の場合GitHubとSSH接続をするところをつまづくと思います。ググってみてください。（ググるよりChatGPTに聞いた方が早いです笑）

~~あとは、以前作ったPortfolioのソースコードを参考に同じようなものを作り直して、再デプロイという流れになります。~~

実際は作り直しというほど大袈裟な話ではなく、以前作ったソースコードをnode\_modulesのディレクトリ以外を移して、`npm run build` コマンドを打てば動くようになりました。私の場合はディレクトリのルートパスが /portfolioになっていたのも、その部分を全て/に置換して、デプロイすればすれば期待通り wecken.github.io でホスティングすることができました。

ここから先は、ホスティング完了後に変更を加える手順です。  
デプロイ用のコマンドを打つのがミソになっていて、間違ってもリモートの gh-pages ブランチを直接変更することがないようにしてください。

# GitHub PagesでホスティングしているReact Appを更新する手順

Reactアプリケーションに変更を加えた後、その更新をGitHub Pagesにデブローイする手順は以下の通りです。このプロセスは、初回デブローイ時に設定したワークフローに基づいています。変更を加え、テストを行い、そして変更を公開する一連のステップを踏みます。

## 1. ローカルでの変更

- 変更の実装:** ローカル環境でReactアプリケーションに必要な変更を加えます。
- ローカルでのテスト:** 変更を加えた後、アプリケーションが正しく動作するかローカルでテストします。これは、アプリケーションの品質を保つために重要です。

```
npm start
```

このコマンドにより、開発サーバーが起動し、変更をリアルタイムで確認できます。

## 2. プロジェクトのビルド

変更が完了し、ローカルでのテストが成功したら、プロダクション用にアプリケーションをビルドします。

```
npm run build
```

このコマンドは\*\*build\*\*ディレクトリにアプリケーションのビルド済みバージョンを生成します。

## 3. GitHubへの変更のプッシュ

ローカルでの変更をGitHubリポジトリにプッシュします。まず、変更をステージングし、コミットします。

```
git add .
git commit -m "変更の説明"
```

そして、リポジトリにプッシュします。

```
git push origin master
```

ここで**\*\*masterはあなたが使用しているメインのブランチの名前です**。main\*\*ブランチを使用している場合は、それに応じてコマンドを調整してください。

## 4. GitHub Pagesへのデプロイ

最後に、変更をGitHub Pagesにデプロイします。これは、以前に設定した**\*\*deploy\*\***スクリプトを実行することで行います。

```
npm run deploy
```

このステップは、ビルドされたアプリケーションを**\*\*gh-pages\*\***ブランチにプッシュし、GitHub Pagesがこのブランチからアプリケーションを自動的に公開するようにします。

## 5. 更新の確認

デプロイ後、GitHub PagesのURLにアクセスして、変更が正しく反映されているかを確認します。変更がすぐには反映されない場合がありますので、少し時間を置いてから再度確認してみてください。

これらの手順を通じて、Reactアプリケーションに加えた変更を簡単に公開できます。定期的な更新や改善を行うことで、ユーザーにとってより良いアプリケーション体験を提供できるようになります。

# おわり



以上です。PortfolioをReactで作ることによって、Figmaなどでのデザイン以上に実際のモノづくりの部分（＝ソフトウェアエンジニアリングの領域）にもこだわっていきたいという思いを表現している部分でもあります。

今回の記事は、自分への戒めとして書いているものなので、誰の役に立つかはわかりませんが、やってみたい人はやってみてください！