

Hosting a Blog on GitHub Pages – Jekyll & React (Vite)

1. Introduction

GitHub Pages is a free hosting service provided by GitHub. It allows you to host static websites directly from a GitHub repository. Two popular options:

- **Jekyll** – Static site generator supported natively by GitHub Pages.
- **React (Vite)** – Modern frontend framework that needs to be built and deployed to GitHub Pages.

This guide covers **setup, configs, issues faced, and fixes** during migration from Jekyll to React.

2. Environment Setup

Install Node.js & npm

1. Download the latest **Node.js LTS** from nodejs.org.
2. During installation, add Node.js to **PATH**.
3. Verify installation:

```
node -v  
npm -v
```

Fix PowerShell Execution Policy (npm.ps1 issue)

If you see:

```
npm : File F:\npm.ps1 cannot be loaded because running scripts is disabled
```

Run PowerShell as Admin and execute:

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

VS Code Setup

- Install **Prettier - Code Formatter** extension.
 - Go to Settings → Search *default formatter* → set **Prettier**.
 - Enable **Format on Save**.
-

3. Hosting with Jekyll on GitHub Pages

Setup

1. Create a new GitHub repo.
2. Push Jekyll project files (including `_config.yml`).
3. Go to **Settings** → **Pages** → **Source** and select `main` branch /docs.
4. GitHub will automatically build the site using Jekyll.

Issues Faced & Fixes

- **Custom CSS not applied** → Ensure SCSS is inside `assets/css` and referenced in `_config.yml`.
 - **GitHub Pages build errors** → Check logs in repo → `Actions` tab → Fix `_config.yml` syntax.
-

4. Hosting with React (Vite) on GitHub Pages

Step 1: Create React Project with Vite

```
npm create vite@latest my-blog
cd my-blog
npm install
npm run dev
```

Step 2: Add React Router

```
npm install react-router-dom
```

Setup in `App.jsx`:

```
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
```

Step 3: Configure for GitHub Pages

- In `vite.config.js` add:

```
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';

export default defineConfig({
  plugins: [react()],
```

```
base: '/<your-repo-name>/',
});
```

- In `package.json` add deploy scripts:

```
"scripts": {
  "dev": "vite",
  "build": "vite build",
  "preview": "vite preview",
  "deploy": "gh-pages -d dist"
}
```

Step 4: Deploy to GitHub Pages

```
npm install gh-pages --save-dev
npm run build
npm run deploy
```

This pushes the `dist` folder to the `gh-pages` branch.

Go to **GitHub Repo** → **Settings** → **Pages** → **Branch: gh-pages**.

Step 5: Common Issues

- **Navigation not working** → Use `react-router-dom` with `Link` instead of `<a>` tags.
- **CSS not loading** → Import CSS inside component: `import './Navbar.css';`
- **Only URL updates, page doesn't change** → Ensure you use `Routes` + `Route` from React Router instead of plain HTML.

5. GitHub Setup Steps (Detailed)

Repository Creation

1. Log in to GitHub → Click **New Repository**.
2. Enter repo name (e.g., `my-blog`).
3. Choose visibility:
4. **Public** → Works with free GitHub Pages hosting.
5. **Private** → Requires GitHub Pro/Team.
6. Do **not** initialize with README if pushing an existing project.
7. Click **Create Repository**.

Local Git Setup & Push

```
git init
git remote add origin https://github.com/<username>/<repo>.git
git add .
git commit -m "Initial commit"
git branch -M main
git push -u origin main
```

GitHub Pages Settings

1. Go to **Repo** → **Settings** → **Pages**.
2. Under **Source**, select branch:
3. For **Jekyll** → `main` branch with `/docs` folder.
4. For **React** → `gh-pages` branch (created by `gh-pages` package).
5. Click **Save**.
6. GitHub will display your site URL (`https://<username>.github.io/<repo>`).

Additional GitHub Configurations

- **Enforce HTTPS** → Ensure “Enforce HTTPS” is checked.
- **Custom Domain** (optional) → Add your domain under Pages settings, then configure DNS (CNAME file will be auto-created).
- **Actions/Build Logs** → If something fails, check **Actions tab** for detailed logs.

Branch Management

- **main** → keep source code.
- **gh-pages** → used only for deployment output.

Common GitHub Issues

- **Site not showing after deploy** → Recheck Pages settings, branch selection, and repo visibility.
- **CSS/JS 404 errors** → Ensure `vite.config.js` has correct `base` path.
- **Private repo not loading** → Free GitHub Pages only works with **public repos**.
- **gh-pages branch not created** → Ensure `gh-pages` package is installed and `npm run deploy` is executed successfully.
- **Conflicts between Jekyll & React** → Remove `_config.yml` when migrating to React to avoid Jekyll processing errors.

6. Conclusion

- Jekyll is supported out-of-the-box but limited.
- React (Vite) gives flexibility and modern stack, but needs extra setup.

- GitHub setup is critical: correct repo visibility, branch configuration, and Pages settings are mandatory.
- Common issues mostly come from misconfigured paths, CSS imports, or GitHub Pages branch settings.



Now you can host either Jekyll or React projects on GitHub Pages successfully!