



React and Vite App Deployment with GitHub Pages



Issue Description

GET https://username.github.io/src/main.jsx net::ERR_ABORTED 404

What this means:

- The `index.html` file is trying to load `src/main.jsx` as if it's a file on the server.
- On **production (GitHub Pages)**, it should load the **bundled and built files**, not the raw `.jsx` files.
- Currently, **source code** is being deployed, but GitHub Pages requires the **built files** from the `dist` folder.



How to Fix It (Step-by-Step Guide)

Step 1: Install `gh-pages`

In the terminal, run the following command inside the project root directory:

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

Step 2: Update `vite.config.js`

Ensure the `base` property in `vite.config.js` points to the GitHub repository name.

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

export default defineConfig({
  base: "/repository-name/", // 🖱️ Replace with the repo name!
  plugins: [react()],
});
```

Step 3: Edit `package.json` Scripts

In the `package.json` file, update the scripts section to include deploy commands:

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

✓ Link the Local Project to GitHub Repository

Step 4: Set the Correct Remote Repository

Inside the project directory, run the following command to link the repository:

```
git remote set-url origin https://github.com/username/repository-name.git
```

Verify with:

```
git remote -v
```

Expected output:

```
origin  https://github.com/username/repository-name.git (fetch)  
origin  https://github.com/username/repository-name.git (push)
```

Step 5: Commit and Push the Code (Optional but Recommended)

If there are changes, commit and push them to GitHub:

```
git add .  
git commit -m "Updated site for gh-pages deployment"  
git push origin main
```

(Replace `main` with the branch name if different)

✓ Deploy to GitHub Pages

Step 6: Ensure Correct Deployment Settings

a) Inside `package.json` :

Make sure the deploy script points to the repository.

```
"deploy": "gh-pages -d dist -r https://github.com/username/repository-name.git"
```

b) Or run the deploy command manually:

```
npx gh-pages -d dist -r https://github.com/username/repository-name.git
```

Step 7: Deploy the Project

Run the deployment command:

```
npm run deploy
```

✓ What happens:

- Builds the app into the `/dist` folder.
- Pushes the `dist` contents to the `gh-pages` branch.
- GitHub Pages will serve the **built files**, **not** the raw source files.

✓ Configure GitHub Pages Settings

Step 8: Select Deployment Source in GitHub

1. Go to the repository:

👉 <https://github.com/username/repository-name>

2. Navigate to:

Settings → **Pages**

3. Under **Source**:

- Select **Branch**: `gh-pages`
- Select **Folder**: `/` (root)

Step 9: Access the Live Site

After ~1 minute, the website should be live at:

👉 `https://username.github.io/repository-name/`

✅ Recap of the Deployment Process

◆ What was wrong?

The project was pointing to `src/main.jsx`, which doesn't exist in the built files. GitHub Pages serves from `/dist/index.html` and needs bundled files.

◆ What was done?

- Built the project with `vite build`.
- Deployed the `/dist` folder with `gh-pages`.

✅ What Not to Do

❌ Don'ts

- ❌ Don't deploy raw `src` files to GitHub Pages
- ❌ Don't manually upload files to GitHub Pages
- ❌ Don't serve `main.jsx` —use the **built** files!

✅ Quick Commands Summary

Action	Command
Install	<code>npm install gh-pages --save-dev</code>
Build & Deploy	<code>npm run deploy</code>
GitHub Pages	Configure GitHub Pages to use the <code>gh-pages</code> branch

Final Result

👉 The website should be successfully live at:
<https://username.github.io/repository-name/>