

# 프로젝트

# GitHub 배포 방법





## ❖ 배포 전 주의 사항

- 메인 화면이나 다른 곳에 Logo 이미지를 import하지 않고 그냥 사용한 경우 import로 이미지를 변경하고 배포하기를 권장한다.

```
import tivingLogo from '../../img/1695032536_1.svg';
import backimg from '../../img/6u5x4l6u5x4l6u5x.png';

export default function Header() {
  return (
    <>
      <div className="header-container">
        <div className="top-box">
          <ul className="tbox01">
            <li>
              <Link to="/">
                <img src={tivingLogo} alt="tving logo" />
              </Link>
            </li>
          </ul>
        </div>
      </div>
    </>
  );
}
```



## ❖ 배포 전 주의 사항

- React Router 사용 시 추가 설정
- 만약 react-router-dom을 사용한다면, BrowserRouter 대신 HashRouter 로 변경해야 한다.
- GitHub Pages는 브라우저 히스토리 라우팅을 직접 처리하지 않기 때문이다.
- import { HashRouter , Routes, Route } from "react-router-dom";
- **바꾸지 않으면 페이지 새로고침 시 404가 나올 수도 있다.**

```
import { HashRouter, BrowserRouter, Routes, Route } from 'react-router-dom';
```

```
function App() {
  return (
    <HashRouter>
      <Header />
      <Routes>
        <Route path="/" element={<Section />} />
        <Route path="/LoginPages" element={<LoginPages />} />
      </Routes>
      <Footer />
    </HashRouter>
  );
}
```

```
export default App;
```



## ① vite.config.js 설정 수정

- vite.config.js 파일을 열어서 base 경로를 설정해야 한다.
- 이건 GitHub 저장소 이름과 같아야 한다.
- 예를 들어 GitHub repo가 <https://github.com/ohseunghyun/my-react-app>
- `base: '/my-react-app/'`,
- 이렇게 꼭 써야 함.

```
vite.config.js > ...
  import { defineConfig } from 'vite';
  import react from '@vitejs/plugin-react';

  // https://vite.dev/config/
  export default defineConfig({
    plugins: [react()],
    base: '/tiving/',
  });
}
```



## ② gh-pages 패키지 설치

- gh-pages 설치
- 터미널에서 아래 명령 실행
- npm install gh-pages --save-dev

```
Admin@DESKTOP-6ITD093 MINGW64 ~/Desktop/tving (main)
● $ npm install gh-pages --save-dev

added 44 packages, and audited 202 packages in 4s
40 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```



### ③ Package.json 설정

package.json의 "scripts" 부분에 다음 줄을 추가하세요.

```
"scripts": {  
  "dev": "vite",  
  "build": "vite build",  
  "preview": "vite preview",  
  "deploy": "gh-pages -d dist" // ← 이 줄 추가  
}
```

- Vite는 npm run build를 하면 결과물이 dist 폴더에 생성된다.
- Gh-pages 패키지가 그 폴더를 자동으로 GitHub Pages에 올려준다.

package.json > {} dependencies

```
{  
  "name": "tving",  
  "private": true,  
  "version": "0.0.0",  
  "type": "module",  
  ▷디버그  
  "scripts": {  
    "dev": "vite",  
    "build": "vite build",  
    "lint": "eslint .",  
    "preview": "vite preview",  
    "deploy": "gh-pages -d dist"  
  }
```



## ④ 배포 실행

- npm run build
- npm run deploy
- npm run build : dist 폴더 생성
- npm run deploy : dist를 gh-pages 브랜치로 푸시
- (단, 프로젝트 수정 후 git업로드 하고, 다시 npm run build, npm run deploy를 진행 해야 함을 주의하자)

```
Admin@DESKTOP-6ITD093 MINGW64 ~/Desktop/tving (main)
$ npm run build

> tving@0.0.0 build
> vite build

vite v7.2.2 building client environment for production...
✓ 52 modules transformed.

dist/index.html          0.47 kB | gzip: 0.29 kB
dist/assets/react-CHdo91hT.svg 4.13 kB | gzip: 2.05 kB
dist/assets/index-CoZwuVdr.css 3.26 kB | gzip: 0.80 kB
dist/assets/index-D-PpLmeO.js  233.07 kB | gzip: 74.33 kB
✓ built in 1.20s
```

```
Admin@DESKTOP-6ITD093 MINGW64 ~/Desktop/tving (main)
$ npm run deploy

> tving@0.0.0 deploy
> gh-pages -d dist

Published
```



## ⑤ 배포 확인

- GitHub Pages 활성화
- GitHub repo → Settings
- 왼쪽 메뉴 → Pages
- Branch 부분을 gh-pages 로 선택
- 저장하면 잠시 후 배포됨

➤ 이미 배포된 상태임 => <https://ohseung-web.github.io/tiving/>

The screenshot shows the GitHub Pages settings page for a repository. On the left, there's a sidebar with sections like General, Access, Collaborators, Moderation options, Code and automation (with branches, tags, rules, actions, models, webhooks, copilot, environments, codespaces, and pages), and a Preview button. The main area is titled "GitHub Pages" and contains a message stating "Your site is live at <https://ohseung-web.github.io/tiving/>". It also shows "Last deployed by ohseung-web 5 minutes ago" and buttons for "Visit site" and "Unpublish site". Below this, under "Build and deployment", there's a "Source" section with a dropdown set to "Deploy from a branch" and a "Branch" section where it says "Your GitHub Pages site is currently being built from the gh-pages branch". A link to "Learn more about configuring the publishing source for your site" is provided. At the bottom of this section, there are dropdowns for "gh-pages" and "/(root)" and a "Save" button, which is highlighted with a red rectangle. At the very bottom, there's a note about the last deployment and a link to "Learn more about deploying to GitHub Pages using custom workflows".