

리액트앱 gitPage에 배포하기

▼ 준비 사항

항목	확인
GitHub 계정	필요
로컬에 Git 설치	필요 (Git 설치 링크)
<code>npm</code> 설치됨	Node.js 설치 시 포함 (Node.js 설치 링크)
React 프로젝트 생성 완료	(예: <code>create-react-app my-app</code> 등)

▼ 프로젝트 배포 단계

1단계: GitHub 저장소 만들기

1. <https://github.com> 접속
2. 오른쪽 위 `+` → **New repository**
3. 저장소 이름을 예: `my-react-app` 입력
4. `[v]` Public 선택
5. `[v] Initialize with a README`는 체크 안 함!
6. `Create repository` 클릭

2단계: 로컬 프로젝트와 GitHub 연결

터미널(cmd)에서 React 프로젝트 폴더로 이동:

```
cd C:\rwork\프로젝트명
```

git 초기화하고 GitHub 연결:

```
git init
git remote add origin https://github.com/계정명/원격저장소명.git
```

예시:

```
git remote add rapp https://github.com/ikarosala7/rapp11.git
```

3단계: **gh-pages** 패키지 설치

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

- GitHub Pages로 자동 배포를 도와주는 도구

4단계: **package.json** 수정

package.json 파일을 열고 아래 2가지를 추가/수정할 것

1. **homepage** 속성 추가 (파일 맨 위에 가까운 곳):

```
"homepage": "https://ikarosala7.github.io/rapp11",
```

! 여기서 yourname, my-react-app은 본인 GitHub 계정과 저장소 이름으로 바꿔야 함.

2. **scripts** 항목 수정 (아래 2줄 추가):

```
"scripts": {  
  "predeploy": "npm run build",  
  "deploy": "gh-pages -d build",  
  ...  
}
```

- **predeploy**: 먼저 build를 실행
- **deploy**: **build** 폴더 내용을 gh-pages 브랜치로 배포

5단계: GitHub에 코드 푸시

```
git config user.email "git이메일주소"  
git config user.name "git계정명"
```

```
git add .
git commit -m "🚀 최초 커밋"

npm run build

# 빌드된 폴더를 gh-pages 브랜치로 강제로 푸시
git checkout --orphan gh-pages
git --work-tree build add --all
git --work-tree build commit -m "Deploy"
git push raoo HEAD:gh-pages --force
git checkout main
```

6단계: 배포 실행

```
npm run deploy
```

- 빌드하고, `gh-pages` 브랜치에 자동으로 올려줌.

7단계: GitHub Pages 활성화 확인

1. GitHub 저장소로 이동
2. 상단 `Settings` 클릭
3. 왼쪽 메뉴에서 **"Pages"** 또는 **"Pages > Deployments"** 클릭
4. 배포 경로가 표시되며, `https://사용자계정명.github.io/레포지토리명/` 로 접속 가능
5. 접속되면 배포 성공!

