



# Deploy

## 배포 준비하기

### 파일 빌드하기

지금까지 우리가 개발 서버에서 사용한 파일들은 루트 경로의 `src` 와 `public` 폴더 안에 있는 파일들이었습니다. 개발 서버는 실시간으로 코드 변화를 감지해 컴파일해 브라우저에 렌더링하는 기능을 가지고 있기 때문에 매우 편리하지만, 실제 서비스로 사용하기에는 성능 면에서 적절하지 않습니다.

실제 프로덕션 환경에서는 자바스크립트와 CSS 파일을 최적화하는 Build 단계를 거치게 되는데, 이를 Create-react-app으로 리액트 앱을 만들 때 같이 설치되었던 `webpack` 이 수행하게 됩니다. 터미널에서 아래 명령어를 입력해 현재 소스 코드를 빌드할 수 있습니다.

```
npm run build
```

### 노드 서버 세팅하기

다음으로는 Node API 서버 프레임워크로 유명한 `express` 를 비롯한 패키지들을 설치해줍니다. 이전에 글로벌로 설치했던 `json-server` 도 배포에 필요하기 때문에 로컬로 다시 한 번 설치합니다.

```
npm install json-server express path dotenv
```

그 다음 express server에서 구동하는데 필요한 `server.js` 파일을 만들어줍니다.

```
const jsonServer = require('json-server');
const app = jsonServer.create();
const path = require('path');
const express = require('express');
const middlewares = jsonServer.defaults();
const router = jsonServer.router('src/db/data.json');

require("dotenv").config();
const port = process.env.PORT;
```

```

app.use('/api', middlewares, router);
app.use(express.static(path.join(__dirname, 'build')));

app.get('/*', function (req, res) {
  res.sendFile(path.join(__dirname, 'build', 'index.html'));
});

app.listen(port);
console.log(`Start server with port ${port}`)

```

위 코드에서 `process.env` 가 환경변수를 읽어오는 부분인데, 이를 위해서 `.env` 파일을 루트 경로에 아래와 같이 만들어줍니다.

```

PORT=3000
REACT_APP_BACKEND_API_URI=/api

```

환경 변수를 리액트에서도 사용할 수 있도록 하려면, `REACT_APP` 이라는 prefix를 달아주어야 합니다. 이제 이 변수를 리액트에서 읽어오도록 `index.js` 에 아래 두 라인을 추가합니다.

```

import React from "react";
import ReactDOM from "react-dom";
import "./index.css";
import App from "./App";
import dotenv from "dotenv"; // HERE

dotenv.config(); // HERE

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById("root")
);

```

이제 다른 컴포넌트에서 기존에 `localhost:3001` 으로 되어있던 경로를 모두 아래와 같이 교체합니다.

```
`${process.env.REACT_APP_BACKEND_API_URI}/items/${todo.id}`
```

그 다음 express server를 실행해 build 환경이 제대로 작동하는지 `localhost:3001` 에서 체크할 수 있습니다.

```
node server.js
```

## Heroku에 배포하기

수업의 마지막 단계로 우리가 만든 리액트 프론트엔드와 `json-server` 백엔드를 배포해 보겠습니다. 무료로 배포하기 위해서 Heroku라는 서버 플랫폼(정확히는 클라우드 컴퓨팅 플랫폼)을 사용할 것입니다.

### Cloud Application Platform | Heroku

Whether you're building a simple prototype or a business-critical product, Heroku's fully-managed platform gives you the simplest path to delivering apps quickly. Learn more about how Heroku

 <https://www.heroku.com/home>



아직 계정이 없다면 회원가입을 합니다.

## 배포에 필요한 파일 준비

`Procfile` 이라는 파일을 루트 경로에 하나 생성하고, 아래와 같이 입력합니다. 여기서 `web` 은 실행할 프로세스의 타입을 의미하는데, 우리는 웹 어플리케이션을 배포하고 있어서 `web` 타입으로 만들고, `:` 뒤에는 실행할 명령어를 입력합니다. 따라서 노드 익스프레스 서버를 실행하는 명령어를 적어줍니다.

```
web: node server.js
```

위와 같이 코드를 수정한 다음, 깃허브 저장소에 업로드합니다.

## Github 소스코드로 배포하기

이제 Heroku에서 `Create new app` 을 이용해 새로운 앱을 생성합니다.

You don't have any apps yet  
Every app and pipeline you create or become a collaborator on will appear here

Create new app

---

Looking for help getting started?  
Get started by reading one of our language guides in the Dev Center

Choose a language guide... ▾

`App name` 은 해당 앱이 실행될 URL 주소입니다. 예를 들어 `App name` 이 `indo-react-todo-list` 라면, 다음과 같이 URL이 만들어집니다.

```
https://indo-react-todo-list.herokuapp.com/
```

App name

indo-react-todo-list



indo-react-todo-list is available

Choose a region



United States



Add to pipeline...

Create app

다음으로 우리는 소스코드를 깃허브에 올려놓았기 때문에 `Deployment method` 를 Github를 선택하고, 밑에서 깃허브에 연결한 다음 저장소를 선택합니다.

## Deployment method



Heroku Git  
Use Heroku CLI



GitHub  
Connect to GitHub



Container Registry  
Use Heroku CLI

### Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

#### Search for a repository to connect to



Indosaram



react-todo-list

Search

Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)



Indosaram/react-todo-list

Connect

배포 방법은 자동과 수동이 있는데, 여기서는 수동을 선택하고 **main** 브랜치를 선택한 다음 **Deploy Branch** 를 클릭합니다.

### Manual deploy

Deploy the current state of a branch to this app.

#### Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

#### Choose a branch to deploy



main



Deploy Branch

빌드 로그가 출력되고, 잠시 후에 배포된 앱을 확인할 수 있습니다.