

13. Building & Deploying React Apps

Last updated by | Subramanya Dixit | May 5, 2025 at 7:59 PM GMT+5:30

Why Building and Deploying React Apps Is Important

Once you've developed your React app, it's crucial to build it for production to optimize performance. Deploying ensures that your app is accessible to users. This topic covers the steps to create a production build and deploy React applications to popular platforms.

Key Concepts

Building a Production-Ready React App

React provides a built-in tool called `Create React App` that allows you to easily create optimized production builds of your app.

Steps to build:

1. Install dependencies:

```
npm install
```



2. Build the app for production:

```
npm run build
```



This creates a `build/` directory with optimized files for production.

3. You can now serve these files using a static file server or deploy them directly to a hosting platform.

```
npm install -g serve  
serve -s build
```

Deploying React Apps

There are multiple platforms for deploying React apps. Here are some popular ones:

1. GitHub Pages

- Install the `gh-pages` package:

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



- Add deployment scripts in `package.json`:

```
"homepage": "http://yourusername.github.io/your-repository",  
"scripts": {  
  "predeploy": "npm run build",  
  "deploy": "gh-pages -d build"  
}
```



- Deploy using:

```
npm run deploy
```



2. Netlify

- Connect your GitHub repository to Netlify.
- Set the build command to `npm run build` and the publish directory to `build/`.
- Netlify will handle the deployment automatically when you push changes to your GitHub repo.

3. Vercel

- Vercel is another popular platform for deploying React apps. Connect your GitHub repo to Vercel and it will automatically deploy your app with each push.

✅ Continuous Deployment (CD)

Many modern platforms like Netlify, Vercel, and GitHub Pages support **Continuous Deployment**. This means your app is automatically built and deployed every time you push code to your repository.

💡 Guidelines

- Always test your app locally before deploying it to production.
- Minimize assets (like images and JavaScript files) for faster load times in production.
- Use environment variables to manage different configurations (e.g., API keys, environment settings).

📄 Practice Exercises

1. Build your app for production and deploy it to GitHub Pages.
2. Set up a deployment pipeline using Netlify or Vercel.
3. Optimize your build by removing unused dependencies and reducing bundle size.

? Quiz Questions

1. How do you build a production-ready React app?

- a) `npm start`
- ✅ b) `npm run build`
- c) `npm install`
- d) `npm deploy`

2. Which command is used to deploy to GitHub Pages?

- a) `npm run deploy`
- b) `npm deploy`
- ✅ c) `gh-pages -d build`
- d) `npm run github`

3. What does Continuous Deployment (CD) do?

- a) Automatically commits code to the repository
 - b) Automatically pushes changes to the production server
 - ☒ **c) Automatically builds and deploys the app on each commit**
 - d) Automatically merges code into the main branch
-