

08

Manually Create a React Project

👉 babel

```
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>  
<script type="text/babel" data-type="module">  
  import React from "https://esm.sh/react@19/?dev";  
  import ReactDOMClient from "https://esm.sh/react-dom@19/client?dev";
```

👉 react

👉 react-dom

```
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>  
<script type="text/babel" data-type="module">  
  import React from "https://esm.sh/react@19/?dev";  
  import ReactDOMClient from "https://esm.sh/react-dom@19/client?dev";
```

👉 the page's performance depends on the network speed

Why Manual Setup?

👉 Modern frameworks pre-install many unnecessary dependencies for now, we will learn React frameworks in the future courses

👉 Eliminating the need to fetch files over the network, thus achieving faster project execution

👉 Manual setup teaches the underlying structure and purpose of each component in the project

Package Managers

👉 Manage dependencies in React projects

👉 npm, Yarn, pnpm, Bun, etc

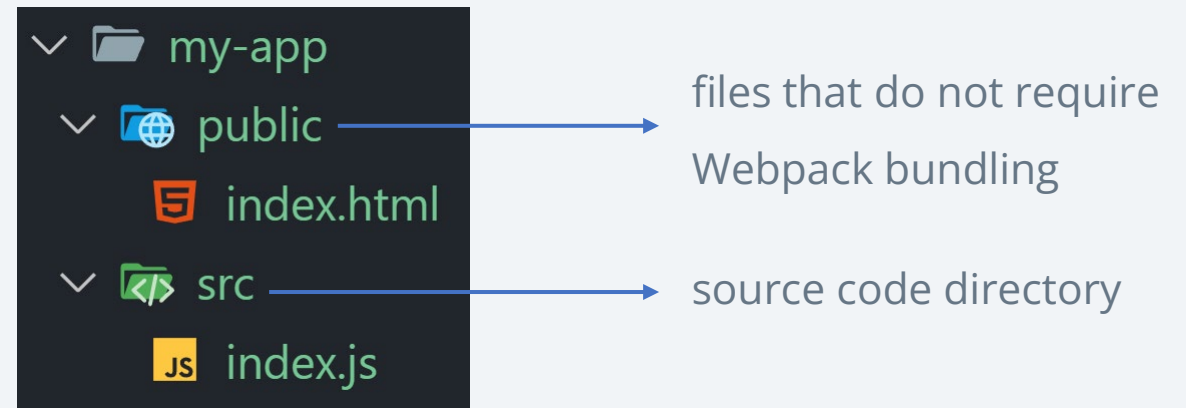
👉 Projects require bundling before running in the browser.

Webpack

- 👉 Bundles files for browser compatibility.
- 👉 Provides a development server for real-time updates
- 👉 Requires extensive configuration

react-scripts

- 👉 Pre-configured environment for React projects
- 👉 Includes Webpack, Babel, testing frameworks
- 👉 "Convention over Configuration."



Creating the Project

Step1. Create the project root folder

Step2. Create `public/index.html` and `src/index.js`

Step3. Initialize the project

- `npm init -y` or `yarn init -y`

Step4. Install dependencies

- `npm i react react-dom react-scripts`

or

- `yarn add react react-dom react-scripts`

Running the Project

👉 Command: `npx react-scripts start`

👉 Real-time updates via hot reloading

👉 Webpack handles the connection between `index.html` and `index.js`

Bundling the Project

- 👉 Command: `npx react-scripts build`
- 👉 Generates a build folder with bundled files
- 👉 Fix local path issues by modifying file paths

Adding Scripts to package.json

👉 Add scripts:

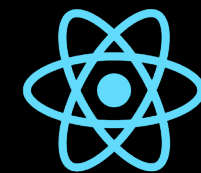
"start": react-scripts start

"build": react-scripts build

👉 Run commands:

- npm start (or npm run start)

- npm run build



Thanks for watching