

Installation

This lesson explains how to get started with building a React application. You will also learn about what CDN is how you can add React to your application through CDN.

There are many approaches to getting started with a React application. The first we'll explore is a CDN, short for [content delivery network](#). Don't worry too much about CDNs now, because you will not use them in this course, but it makes sense to explain them briefly. Many companies use CDNs to host files publicly for their consumers. Some of these files are libraries like React, since the bundled React library is just a *react.js* JavaScript file.

To get started with React by using a CDN, find the `<script>` tag in your web page HTML that points to a CDN URL. You will need two libraries: *react* and *react-dom*.

```
<script
  crossorigin
  src="https://unpkg.com/react@16/umd/react.development.js"
></script>

<script
  crossorigin
  src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"
></script>
```

You can also get React into your application by initializing it as node project. With a *package.json* file, you can install *react* and *react-dom* from the command line. However, the folder must be initialized as a npm project using `npm init -y` with a *package.json* file. You can install multiple node packages with npm:

```
npm install react react-dom
```

This approach is often used to add React to an existing application managed with npm.

You may also have to deal with [Babel](#) to make your application aware of JSX (the React syntax) and JavaScript ES6. Babel transpiles your code—that is, it converts it to vanilla JavaScript—so most modern browsers can interpret JavaScript ES6 and JSX. Because of this difficult setup, Facebook introduced *create-react-app* as a zero-configuration React solution. The next section will show you how to set up your application using this bootstrapping tool.

Further Reading:

- Read about [React installations](#)