



Create a New React App

Use an integrated toolchain for the best user and developer experience.

This page describes a few popular React toolchains which help with tasks like:

- Scaling to many files and components.
- Using third-party libraries from npm.
- Detecting common mistakes early.
- Live-editing CSS and JS in development.
- Optimizing the output for production.

The toolchains recommended on this page **don't require configuration to get started**.

You Might Not Need a Toolchain

If you don't experience the problems described above or don't feel comfortable using JavaScript tools yet, consider adding React as a plain `<script>` tag on an HTML page, optionally with JSX.

This is also **the easiest way to integrate React into an existing website**. You can always add a larger toolchain if you find it helpful!

Recommended Toolchains

The React team primarily recommends these solutions:



- If you're **learning React** or **creating a new single-page app**, use [Create React App](#).
- If you're building a **server-rendered website with Node.js**, try [Next.js](#).
- If you're building a **static content-oriented website**, try [Gatsby](#).
- If you're building a **component library** or **integrating with an existing codebase**, try [More Flexible Toolchains](#).

Create React App

[Create React App](#) is a comfortable environment for **learning React**, and is the best way to start building a **new single-page application** in React.

It sets up your development environment so that you can use the latest JavaScript features, provides a nice developer experience, and optimizes your app for production. You'll need to have Node ≥ 6 and npm ≥ 5.2 on your machine. To create a project, run:

```
npx create-react-app my-app
cd my-app
npm start
```

Note

npx on the first line is not a typo — it's a [package runner tool that comes with npm 5.2+](#).

Create React App doesn't handle backend logic or databases; it just creates a frontend build pipeline, so you can use it with any backend you want. Under the hood, it uses [Babel](#) and [webpack](#), but you don't need to know anything about them.

When you're ready to deploy to production, running `npm run build` will create an optimized build of your app in the `build` folder. You can learn more about Create React App [from its README](#) and the [User Guide](#).

Next.js

Next.js is a popular and lightweight framework for **static and server-rendered applications** built with React. It includes **styling and routing solutions** out of the box, and assumes that you're using [Node.js](#) as the server environment.

Learn Next.js from [its official guide](#).

Gatsby

Gatsby is the best way to create **static websites** with React. It lets you use React components, but outputs pre-rendered HTML and CSS to guarantee the fastest load time.

Learn Gatsby from [its official guide](#) and a [gallery of starter kits](#).

More Flexible Toolchains

The following toolchains offer more flexibility and choice. We recommend them to more experienced users:

- **Neutrino** combines the power of [webpack](#) with the simplicity of presets, and includes a preset for [React apps](#) and [React components](#).
- **nwb** is particularly great for [publishing React components for npm](#). It [can be used](#) for creating React apps, too.
- **Parcel** is a fast, zero configuration web application bundler that [works with React](#).
- **Razzle** is a server-rendering framework that doesn't require any configuration, but offers more flexibility than Next.js.

Creating a Toolchain from Scratch

A JavaScript build toolchain typically consists of:

- A **package manager**, such as [Yarn](#) or [npm](#). It lets you take advantage of a vast ecosystem of third-party packages, and easily install or update them.
- A **bundler**, such as [webpack](#) or [Parcel](#). It lets you write modular code and bundle it together into small packages to optimize load time.

- A **compiler** such as [Babel](#). It lets you write modern JavaScript code that still works in older browsers.

If you prefer to set up your own JavaScript toolchain from scratch, [check out this guide](#) that re-creates some of the Create React App functionality.

Don't forget to ensure your custom toolchain [is correctly set up for production](#).

[Previous article](#)

[Add React to a Website](#)

[Next article](#)

[CDN Links](#)

DOCS

Installation

Main Concepts

Advanced Guides

API Reference

Contributing

FAQ

COMMUNITY

Community Resources

Tools

CHANNELS

GitHub [↗](#)

Stack Overflow [↗](#)

Discussion Forum [↗](#)

Reactiflux Chat [↗](#)

DEV Community [↗](#)

Facebook [↗](#)

Twitter [↗](#)

MORE

Tutorial

Blog

Acknowledgements

React Native [↗](#)

Copyright © 2018 Facebook Inc.