

How can I build a full stack application with React?

Written by Seth Corker on Benevolent Bytes

React is an excellent framework for building user interfaces but it's not clear how you can use React to build an entire full stack application. There are a few different ways to approach this because React doesn't tell you how to do it.

The SPA approach

A very flexible approach to building a fullstack application with React is to define, or use your existing stack. Do you have a Django app written in Python? A dotnet core web api? A Ruby on Rails app? Whatever you have or want to use for the backend is up to you. You can mix and match the technologies you and your team are comfortable with and use React on the frontend as a single-page application or SPA.

This is a great approach with existing infrastructure in place but what if you're starting from scratch? A framework might be a better option to provide a useful structure to frame your React knowledge.

Looking for frameworks

Next.js

A popular framework is Next.js because it's has some great features out of the box. You can use your existing React knowledge to build server-side rendered, statically generated or fully client side pages all within the same application. This flexibility is useful as your application evolves and your needs change. It comes with file-based routing and a way to make API routes for backend logic. It's a relatively light framework that doesn't tell you what database you should use or define a structure for the backend so there are still some decisions to be made.

If you want something a little more opinionated, RedwoodJS might be a better option.

RedwoodJS

RedwoodJS is still in the early stages but it takes a different approach to Next.js. It makes more decisions for you on the types of databases that work out of the box, how to model data defined practices for data fetching. It's invested in the Jamstack philosophy so it leverages other services to perform some tasks like authentication. You have flexibility in choosing which services you want to use but there's a happy path that Redwood makes the developer experience great. Offering generators, easy integrations for auth, file upload and testing.

If you're not convinced and want something new but a little more batteries included, BlitzJS might be for you.

Blitz

Blitz is a new framework built on top of Next.js, it takes the same approach as Ruby on Rails and aims to be the all-in-one solution. It uses Next.js as a foundation and adds everything that you'd need in a production application including auth, data fetching, a data layer, background tasks and generators. If you believe in convention over configuration then Blitz has you covered. It's a batteries included framework that will allow you to build a monolithic application with ease.

Wrapping up

As you can see, there are lots of options to leverage your React knowledge to build full stack applications. I've listed some, there are plenty more. Whether you want a light touch like Next.js or batteries included with Blitz or somewhere in between with RedwoodJS. You can also leverage an existing backend and build a new interface in React. It's a great library for building user interfaces that doesn't prescribe how you should build the rest of your application, this can be challenging at first but if you embrace it you can create a stack that works for you.