

---

# Introduction to React and Drupal

Add to queue

Share

Last updated April 1, 2020

Theming

Module Development

8.9.x/9.0.x

[React](#) is a JavaScript library that makes it easy to create interactive user interfaces. [Drupal](#) is a content management system with a powerful web services API. React and Drupal can work together in a couple of different ways. This series of tutorials explores some of those options, and the related concepts and terminology.

Our goal is to provide you with some baseline information you can use to get started integrating React and Drupal together. After you've completed these tutorials you should be able to better envision how to use these two technologies to solve some of your own issues. We hope you'll come away from this with enough knowledge to start exploring further on your own. We link to external resources liberally, and encourage you to explore beyond the examples we provide, read the linked resource, and dig in. There's no substitute for exploration and experimenting with real code when it comes to learning these things.

In this series we'll:

- Introduce the technical side of React, terminology, and information about where to find more resources
- Learn about using Drupal, and the JSON:API module, to turn Drupal into a powerful web services API provider
- Learn about the use-cases for adding React to a Drupal site

- Walk through increasingly complex code examples that start with a "Hello World!" application, and end with a fully decoupled application that can list, create, update, and delete content in Drupal
- Learn about how to authenticate a Drupal user via an API using React
- Contrast, through example code and use-cases, the differences between integrating React into an existing Drupal theme or module and creating a stand-alone React application
- And more!

## Prerequisites

- [Web Services in Drupal 8](#). This series assumes you're comfortable administering Drupal and using the core JSON:API web service.
- Familiarity with ES6+ JavaScript

## Why React?

[React](#) is an open-source JavaScript library that lets you build interactive front-end interfaces. React has a large ecosystem of packages and tools to extend the base library, as well as an active community of users. And, if you've been following along with either the Drupal community or the front-end developer community, you'll already know that there's a lot of excitement around React, and for good reason.

Did you know that [Drupal is looking to adopt React](#) to create a better user experience for site builders and content creators? Knowing how to use React, and how to integrate it with Drupal, may become an important skill for Drupal developers.

For a high-level overview of React check out the [React basics](#) tutorial before learning to [add React to an existing Drupal theme](#).

## Why Drupal?

[Drupal](#) is a powerful [content management system](#) that makes it easy to create editorial workflows and interfaces that meet your specific needs.

For a high-level overview of Drupal, especially as a backend for React applications, check out [An Introduction to Drupal for React Developers](#).

Then dive in and learn more about configuring Drupal to act as a web services API and how to [Retrieve Data from an API with React](#).

## Goal

Our goal is to introduce you to the fundamental concepts that you'll need to understand when using React and Drupal together. We don't answer every single question you might have (though feel free to ask us your specific questions), or cover every possible way of doing things. Rather, we want to give you enough information about how all the moving parts work so that you'll be able to start experimenting and solving real-world problems ASAP.

The applications we build in these examples aren't the prettiest, or most complete. We've chosen to forgo things like adding lots of CSS and animations to keep things focused on the fundamental concepts. That said, we encourage you to take the code and improve it with CSS, animations, and better error handling. Then share it with us so we can see what you've come up with!

## The tutorials

**React Basics:** In this tutorial we introduce some of the fundamental concepts of React -- including components, state, and ES6 -- and talk about some of the things you'll *eventually* want to learn as you get further into using React.

**Decoupled vs. Progressively Decoupled:** In this tutorial we explore the difference between fully decoupled and progressively decoupled applications. We'll explain what they are, and provide links to resources where you can further explore the advantages and disadvantages of both. Throughout this series we'll start by creating a progressively decoupled application where we use React within the context of an existing Drupal theme in order to provide some extra UX features. Our application will read and write data via Drupal's API. As the code base becomes larger and more complex we'll look at how to create a stand-alone React application. Hopefully, going through the process of writing code that uses both of these methods will help you better understand which will work best for your use-case.

**Connect React to a Drupal Theme or Module:** In this tutorial we'll create a "Hello World" React application and demonstrate how to connect that React JavaScript library, and our custom

JavaScript code, to a Drupal theme using asset libraries. We'll also dive deep into configuring Webpack (or any other JavaScript bundler) to work in the context of a Drupal theme or module.

**Create a React Component:** In this tutorial we'll create a new React component that queries the Drupal.org project API to get data about project usage and then displays it in a block. This is a great example of how you can use React to enhance an existing Drupal theme without having to go *fully decoupled*.

**Retrieve Data from an API with React:** In this tutorial we'll look at how to make HTTP requests with JavaScript to retrieve data from a web services API. Specifically, we'll talk about using the ES6 native Fetch API and `fetch()` function. We'll go over the anatomy of a fetch request, and also discuss alternatives and when they might come in handy.

**Use React to List Content from Drupal:** In this tutorial we create a new component that uses the JavaScript Fetch API to retrieve a list of nodes from Drupal's JSON:API, loop over the items in the list, and display them with React. This is a good example of how you can work with data either from Drupal or an external API to provide unique user experiences.

**Add, Edit, and Delete Drupal Nodes with React:** In this tutorial we lay the groundwork for an application that can list, create, update, and delete nodes via Drupal's API. We'll look at how to make HTTP POST, PUT, and DELETE requests to manipulate data stored in Drupal via React. We'll also discuss how authentication works in this scenario.

**Build an Interface to Edit Nodes with React:** In this tutorial we'll combine everything we've learned up to this point to create a React application that allows a user to create, read, update, and delete content. This version is still embedded into an existing Drupal page. It might not be super practical, but it does a good job of demonstrating how a React application is structured, as well as more complex API requests.

**Create a Fully Decoupled React Application:** In this tutorial we change gears and start work on creating a fully decoupled React application.

**Start with create-react-app:** In this tutorial we'll use `create-react-app` to scaffold a new fully decoupled React codebase.

**Make API requests with OAuth:** In this tutorial we'll learn how to configure the Simple OAuth Drupal module and use a password grant flow to create authenticated API requests.

**Use Fetch and OAuth to Make Authenticated Requests:** In this tutorial we'll add a login form to collect a username and password. Then we'll exchange those for an OAuth access token, and update existing `fetch` requests to use the OAuth access token.

## Example code

The example code for this set of tutorials is at <https://github.com/DrupalizeMe/react-and-drupal-examples/>. It consists of a Drupal 8 site, a Drupal theme with a React application included in it, and a decoupled React application that works with the Drupal 8 backend. You can use them as a reference while following along, or set up and use your own local development environment.

## Recap

This tutorial provided an outline of what you can expect to learn in the rest of the tutorials in this series.

## Further your understanding

- [Decoupling Explained](#)
- List some things on your site that you think could benefit from adding some interactivity to them.
- Are there things about your user experience that would be better without the need for a page refresh?

## Additional resources

- [React](#) (reactjs.org)
- [@reactjs](#) (twitter.com)
- [React Native](#) (facebook.github.io)
- [25 Amazing Open Source React.js Projects for the Past Year \(v.2018\)](#) (medium.mybridge.co)
- [Things nobody will tell you about React.js](#) (medium.com)
- [Drupal looking to adopt React](#) (dri.es)

---

Was this helpful?

Yes

No

---

[Next tutorial](#) ►

## Get Started Using React and Drupal Together

---

1 [Introduction to React and Drupal](#) Free

---

2 [React Basics](#)

---

3 [Decoupled vs. Progressively Decoupled](#)

---

4 [Connect React to a Drupal Theme or Module](#) Free

---

5 [Create a React Component](#)

---

6 [Add Webpack Hot Module Replacement \(HMR\) to a Drupal Theme](#)

---

7 [Retrieve Data from an API with React](#)

---

8 [Use React to List Content from Drupal](#)

---

9 [Create, Update, and Delete Drupal Content with JavaScript](#)

---

10 [Build an Interface to Edit Nodes with React](#)

---

11 [Create a Fully Decoupled React Application](#) Free

---

12 [Use create-react-app to Start a Decoupled React Application](#)

---

13 [Make API Requests with OAuth](#)

---

14 [Use Fetch and OAuth to Make Authenticated Requests](#)

---

[About us](#)

[Blog](#)

[Student discounts](#)

[FAQ](#)

[Support](#)

[Privacy policy](#)

[Terms of use](#)

[Contact us](#)

## STAY INFORMED

Sign up for our mailing list to get Drupal tips and tricks in your inbox!

[Subscribe](#)

## STAY CONNECTED

Powered by:

Drupalize.Me is a service of [Osio Labs](#), © 2020