



Headless Drupal with React: Complete Guide for Modern Web Development

Last Updated On September 04, 2024



Mayur Upadhyay

9 Minutes to read

Table of Contents

1. What is Headless Drupal?
2. What is React?
3. Why Use Headless Drupal with React?
4. How to Use Headless Drupal with React?
5. FAQs About Using Headless Drupal with React
6. Final Words

As the demands for flexibility, scalability, and performance rise, traditional content management systems (CMS) often fall short. That's where the concept of "headless CMS" comes into play.

When headless Drupal is used with React you can separate the frontend layer from the backend content management. That provides web developers a space for building frontend and backend independently.

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy. [Accept](#)

Sub

We're Online!
How may I help you tod...



Headless Drupal is a decoupled approach to Drupal

where the frontend (user interface) is separated from the backend (content management system). In a traditional CMS setup, Drupal handles both the backend and the frontend.

However, in a headless setup, Drupal only takes care of the backend and the frontend is managed by a separate web development technology. The frontend framework could be **JavaScript frameworks** like React, Angular, or Vue.js. This separation allows for greater flexibility and scalability in building websites and applications.

How Does Headless Drupal Work?

To know how to use headless Drupal with React, you need to understand how it works. Here is the working of headless Drupal:

1. Content Creation: Content editors use Drupal's admin interface to create and manage content. This part remains largely unchanged from a traditional Drupal setup.

2. Content Storage: The content is stored in Drupal's database, just like in a traditional CMS setup.

3. API Layer: Drupal exposes the content through APIs (JSON, REST, GraphQL). These APIs are responsible for providing the content to the frontend applications.

4. Frontend Application: A separate frontend application, built using frameworks like React, fetches the content from Drupal's API endpoints and displays it to users. This application can be hosted separately and interact with Drupal purely through HTTP requests.

This working of headless Drupal allows web developers to build dynamic interfaces and robust functionality.

What is React?

React is a popular open-source JavaScript library used for building user interfaces, especially for single-page applications (SPAs). It lets you build complex UIs from smaller, reusable pieces called components. Using its state management system, you can manage components and handle dynamic data.

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy. X

We're Online!
How may I help you tod...



Key Features of React

React offers a robust set of features that make it a popular choice for building sites with interactive frontend. Here are some of its key features:

Component-Based Architecture: React structures applications using components, which are self-contained and reusable pieces of UI. This approach simplifies development and maintenance, as components can be combined to build complex UIs.

Declarative UI: With React's declarative style, developers define what the UI should look like for different website states. React automatically manages the rendering process, ensuring that the UI stays in sync with the application state.

Virtual DOM: It uses a Virtual DOM to efficiently manage updates to the actual DOM. Changes are first made in the virtual representation, and only the necessary updates are applied to the real DOM, optimizing performance.

JSX (JavaScript XML): JSX is a syntax extension that allows developers to write HTML-like code directly in JavaScript. It simplifies component creation by combining UI design and logic, making the code more readable and easier to write.

Unidirectional Data Flow: Data in React flows in one direction—from parent to child components—through props. This predictable data flow makes it easier to manage and debug the application's state.

State Management: React components can maintain their own state, which is used to store data that may change over time. It re-renders the component automatically when the state changes, keeping the UI updated.

React Hooks: Hooks are functions that let developers use state and other React features in functional components. Hooks like useState and

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy. X

We're Online!
How may I help you tod...



applications (SPAs) that handle routing dynamically.

React Native: It allows developers to use React for building mobile applications. React Native leverages React's component-based architecture to create native mobile apps for iOS and Android using a shared codebase.

These features make the frontend development easy and efficient. Using React allows developers to build mobile friendly sites with native development features. That is one of the reasons why web development experts prefer to build interactive and functional websites.

Struggling to leverage the power of headless Drupal?

[Start Here](#)

Why Use Headless Drupal with React?

The combination of Headless Drupal and React offers a powerful and flexible solution for building modern web applications. Here are key reasons why developers might choose this architecture:

Flexibility: Headless Drupal separates the frontend from the backend, allowing developers to choose React or any other frontend technology to build the UI. This decoupling provides greater flexibility in designing and makes it easier to implement changes without affecting the backend system.

Performance: By using React, which utilizes a Virtual DOM, developers can achieve faster and more efficient UI updates. React's component-based rendering and its efficient handling of state changes lead to quicker page loads, enhancing overall performance.

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy. X

We're Online!
How may I help you tod...



complex content models, workflows, and permissions on the backend. Conversely React efficiently manages the rendering and state on the frontend. This separation allows each layer to build independently for higher scalability.

Reusability: React's component-based architecture allows developers to create reusable components. This reusability speeds up the [web development process](#), ensures consistency, and reduces the amount of code that needs to be written.

API-First Approach: Headless Drupal reveals content through APIs, enabling a consistent and centralized content delivery mechanism. React can retrieve these APIs to fetch and display content, ensuring that data is consistent across different platforms and devices.

Multi-Channel Content Delivery: Headless Drupal with React supports delivering content across channels, such as web and IoT devices. Developers can use a single backend to manage content and APIs. That lets them ensure consistent content delivery across various platforms.

These strengths of both Headless Drupal and React allow professional web development services to build fast, scalable, and secure applications. Using these combinations would be ideal for projects requiring complex content management with a dynamic frontend. Now, let's find out how to use both of them together for your project.

How to Use Headless Drupal with React?

Using Headless Drupal with React involves setting up Drupal as the backend to manage content and React as the frontend to present the content. Here's a step-by-step guide on how to integrate Headless Drupal with React:

Prerequisites

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy. X

We're Online!
How may I help you tod...



Familiarity with making HTTP requests and handling APIs in React using libraries like Axios or Fetch API.

Once you have these prerequisites, we can begin the setup process.

1. Set Up Drupal as a Headless CMS

To start using Drupal as a headless CMS, you need to install and configure it to handle content management and expose data via APIs. This involves setting up Drupal, enabling essential modules, and configuring content types.

Step 1: Begin by installing Drupal. Download it from the official Drupal website and follow the installation guide to set up the backend.

Step 2: Enable and activate the required modules like JSON to expose Drupal content as JSON. To do so, go to Extend in the Drupal admin panel, search for JSON:API, and enable it. This module will allow you to access Drupal content through API endpoints.

Step 3: Set up your content types in Drupal. These could include articles, pages, or custom **content types**. Define the necessary fields and configure their settings according to your content needs.

In this step, we installed Drupal and configured it with modules like JSON:API to expose content as JSON. Content types and fields were defined to structure the content.

2. Expose Drupal Data via JSON

This step involves making Drupal's content accessible through API endpoints using the JSON module. Here, we'll expose content in a format that React can consume, facilitating data exchange between the backend and frontend.

Step 1: Once the JSON:API module is enabled, Drupal automatically creates endpoints for each content type. For example, accessing <https://your-drupal-site.com/jsonapi/node/article> will provide a JSON

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy. X

We're Online!
How may I help you tod...



Step 3: Use Views for custom data endpoints if needed.

Go to Structure > Views and add a new view with REST export.

With JSON enabled, you can access Drupal content via predefined endpoints. That allows you to fetch data from Drupal and use it in your React application.

3. Setting Up a React Project

Setting up a React project involves creating a new React application and installing the necessary dependencies. This environment will serve as the frontend where you'll build user interfaces and handle interactions with the Drupal backend.

Step 1: Create a React App using `create-react-app` to initialize your React project:

```
1 | npx create-react-app my-headless-drupal-app
2 | cd my-headless-drupal-app
```

Step 2: Install Axios to handle API requests:

```
1 | npm install axios
```

Step 3: Set up environment variables for your Drupal API endpoint. Create a `.env` file in your React project root:

```
1 | REACT_APP_DRUPAL_API=https://your-drupal-si
```

Here we created a new React project using `create-react-app`, and the necessary dependencies, like `axios`, were installed.

4. Fetching Data from Drupal in React

In this step, you'll create a service in React to fetch data from Drupal's API. This involves writing functions to handle API requests and retrieve content from your Drupal backend for use in your React components. Here is a sample code to create an API service in React:

```
1 | // src/api/drupalApi.js
2 | import axios from 'axios';
3 | const API_BASE_URL = process.env.REACT_APP
4 | export const fetchArticles = async () => {
5 |   try {
```

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy. X

We're Online!
How may I help you tod...



React's state.

5. Displaying Drupal Content in React Components

After fetching data, you'll display it in React components.

This step involves creating components that render content fetched from Drupal, ensuring that the data is presented in a user-friendly format.

Step 1: Use the `fetchArticles` function in your React components to retrieve and display data from Drupal.

Here is how you can write API service in your React components:

```

1 // src/components/ArticlesList.js
2 import React, { useEffect, useState } from
3 import { fetchArticles } from '../api/drupal
4 const ArticlesList = () => {
5   const [articles, setArticles] = useState(
6     []
7   );
8   useEffect(() => {
9     const getArticles = async () => {
10       const data = await fetchArticl
11       setArticles(data.data);
12     };
13     getArticles();
14   }, []);
15   return (
16     <div>
17       <h1>Articles</h1>
18       <ul>
19         {articles.map((article) =>
20           <li key={article.id}>{
21             ))
22           </ul>
23         );
24       }
25       export default ArticlesList;

```

Step 2: Use JavaScript's `map()` function to iterate over the data and render it within the React components:

```

1 const ArticleList = ({ articles }) => (
2   <div>
3     {articles.map(article => (
4       <div key={article.id}>
5         <h2>{article.attributes.ti
6         <p>{article.attributes.bod
7         )
8       </div>
9     )
10   );

```

Here, we created React components and mapped them to display the content dynamically based on the data stored in the state.

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy.

We're Online!
How may I help you tod...



Step 1: To handle navigation within your React application, install React Router:

```
1 | npm install react-router-dom
```

Step 2: Create routes in your React app to navigate between different components or views:

```
1 | // src/App.js
2 | import React from 'react';
3 | import { BrowserRouter as Router, Route, S
4 | import ArticlesList from './components/Art
5 | import ArticleDetail from './components/Ar
6 | const App = () => {
7 |   return (
8 |     <Router>
9 |       <Switch>
10 |         <Route path="/" exact comp
11 |           <Route path="/article/:id"
12 |             </Switch>
13 |           </Router>
14 |         );
15 |       );
16 |       export default App;
```

Installing and configuring React Router lets it handle navigation in your website. Plus, setting up routes to different components, enabling users to move seamlessly through the site.

7. Handling Authentication and Permissions

If your Drupal content requires authentication, configure token-based authentication or OAuth 2.0 in Drupal.

Step 1: Here is how you can install and configure OAuth module to manage user access:

```
1 | composer require drupal/simple_oauth
2 | drush en simple_oauth -y
```

To configure OAuth settings in Drupal admin, navigate to Configuration > People > Simple OAuth.

Step 2: Use Axios or Fetch API to include authentication tokens in your requests. Update your api.js to handle token-based authentication:

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy.

We're Online!
How may I help you tod...



```

4 | export const fetchArticles = async () => {
5 |   try {
6 |     const response = await axios.get(`${AP
7 |       headers: { 'Authorization': `Bearer
8 |         });
9 |       return response.data;
10 |     } catch (error) {
11 |       console.error("Error fetching articles
12 |         throw error;
13 |     }
14 |   };

```

This step involved securing the site by setting up authentication in both Drupal and React to manage access to sensitive data.

8. Adding Interactivity and Performance

To make the site more engaging, add interactivity and optimize user experience. This involves implementing dynamic features and improving the overall responsiveness of the application.

Step 1: Use React's Context API or a state management library like Redux to manage the application's state, especially as it grows in complexity.

Step 2: Consider using libraries like Material-UI or Bootstrap for modern UI components. These libraries provide ready-to-use components that can enhance the look and feel of your site.

Step 3: Optimize performance by implementing techniques like code splitting, lazy loading of components, and efficient data fetching strategies. Tools like React's Suspense and React.lazy can help with this.

Adding interactivity and enhancing performance will provide better responsiveness and user experience.

By integrating Headless Drupal with React, you leverage Drupal's powerful CMS and React's dynamic frontend capabilities. The setup requires expertise in various development technologies which could be complex. So, it's recommended to contact our web development company to build flexible and scalable sites.

FAQs About Using Headless Drupal with React

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy. X

We're Online!
How may I help you tod...



HTTP requests to the specific URLs that are mapped to the content you want to retrieve.

Q 2. How do I fetch data from Drupal in React?

Use axios or a similar HTTP client library to make requests to the Drupal REST API. Parse the JSON response and update React's state.

Q 3. What kind of content can I manage with Headless Drupal?

Headless Drupal can manage a wide range of content, including articles, pages, media, and more. Since Drupal is highly extensible, you can create custom content types and fields to match the needs of your application. That makes it suitable for anything from simple websites to complex content-driven applications.

Final Words

Headless Drupal with React offers a scalable and flexible solution for building modern websites. By decoupling the frontend and backend, you can leverage the strengths of both technologies to create high-performing sites.

The integration process involves setting up Drupal as a headless CMS, building a React frontend, and handling state management. Considerations such as performance optimization, security, and authentication should also be addressed.

If you want to leverage the complete potential of headless to build robust sites, [hire web developers](#).

Want us to build a well optimized website?

[Contact Us](#)

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy. X

We're Online!
How may I help you tod...



Mayur Upadhyay

Mayur Upadhyay is a tech professional with expertise in Shopify, WordPress, Drupal, Frameworks, jQuery, and more. With a proven track record in web development and eCommerce development.

[Previous](#)[Next](#)

Related Blog Posts

Drupal



Drupal

Complete Setup for Drupal Search API Solr: The Complete Setup & Optimization Guide

Drupal's Search API Solr supercharges your site's search by integrating Apache Solr, an enterprise-grade search engine, for blazing speed, advanced filtering, and smarter relevancy. In...

JULY 31, 2025

Mayur Upadhyay



Symfony

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy.

We're Online!
How may I help you tod...



Drupal Symfony Mailer: A Beginner's Guide

The Drupal Symfony Mailer system offers a flexible and modern way to handle email sending in Drupal. From basic configuration to templated messages and third-party...

CALENDAR JULY 28, 2025

PERSON Mayur Upadhyay



Drupal

What is Drupal CDN and How It Improves Your Website Performance

Using a CDN with your Drupal site can dramatically improve speed, performance, and security. It reduces load times, handles traffic spikes, and lowers server stress...

CALENDAR JULY 14, 2025

PERSON Mayur Upadhyay

Technologies

- Custom PHP Development Company
- WordPress Development Company
- Laravel Development Company
- React Native App Development Company
- Drupal Development Services
- Shopify Development Company
- ReactJS Development Company

Services We Offer

- Web Development Company
- eCommerce Development Company
- Mobile App Development Company
- Web Design Company
- Marketplace Development Company
- CMS Development Company
- Website Maintenance

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you X
agree to our a Cookie Policy.

We're Online!
How may I help you tod...



Hire PHP Developers	eCommerce & Retail
Hire Laravel Developers	Healthcare
Hire ReactJS Developers	Media Entertainment
Hire WordPress Developers	Music
Hire Drupal Developers	Business Services
Hire Dedicated Shopify Developers	Social Networking
Hire MERN Stack Developers	Sports Gambling
Hire MEAN Stack Developers	Real Estate
Hire Dedicated Java Developers	
Hire Webflow Developers	

SERVING IN 10+ COUNTRIES FOR WEB AND MOBILE APP DEVELOPMENT

United States (USA), United Kingdom (UK), Canada, Australia, United Arab Emirates (UAE), Singapore, Spain, Germany, Italy, Switzerland, Sweden, Denmark, New Zealand, Philippines, Slovenia, Nigeria, Jordan, Latvia, Ireland, South Africa, Israel, Netherlands

Location

INDIA	USA	CANADA	UK
704, City Center 2, Science City Road, Ahmedabad -380060, India	131 Sweet Alyssum dr Ladson, South Carolina, 29456, USA	17-430 Mapleview Dr E, Barrie, Ontario – L4N0R9, Canada	53 Kensington Lane, London W8 4QH
	Call Us Now	Call Us Now	Call Us Now
	Call Us Now		

Get In Touch

Whatsapp

sales@wpwebinfotech.com

+1 (857) 380 9517

[Download Brochure](#)

Recognition



4.7 ★★★★★



JSS 98%



4.8 ★★★★★

Join 5348 Subscribers

Get exclusive access to new tips, articles & updates on Technology. It's free!

By continuing to use this website you agree to our a Cookie Policy. X

We're Online!
How may I help you tod...

[Sitemap](#) | [About Us](#) | [Careers](#) | [Blog](#)**Join 5348 Subscribers****Get exclusive access to new tips, articles & updates on Technology. It's free!**

By continuing to use this website you X
agree to our [Cookie Policy](#).

We're Online!
How may I help you tod...

