



# Server-side rendering a Drupal site with Next.js

Presented by: **John Albin Wilkins**

December 9, 2024





## Who is **JohnAlbin?**

- Web developer since 1993
- Drupal dev since 2004
- Drupal core dev since 2006
- React dev since 2017
- 70+ presentations and keynotes in 4 continents
- The Next Drupal maintainer that added App Router support







# Server-side rendering

If Drupal can already do it,  
why have Next.js render on the server?



PHP  
rendered on  
the server

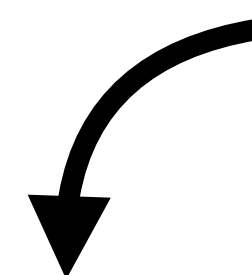


JavaScript  
rendered on  
the server



# Server-side rendering

**0 ms**



HTML  
on  
the server

If Drupal can already do it,  
why have Next.js render on the server?



PHP  
rendered on  
the server

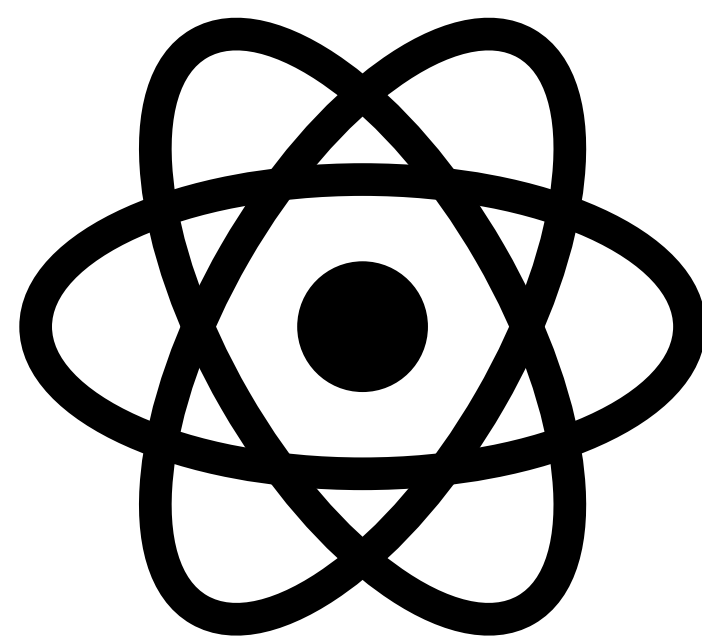


JavaScript  
rendered on  
the server





# React.js Server Components & Client Components



**React.js v19 (pre-release)**

React Server Components

React Client Components



**Next.js v15 (official release)**

App Router



# App Router Terminology

**URL:** `https://example.com/blog/drupalcons/singapore`

**Path:** `https://example.com/blog/drupalcons/singapore`

**Path segments:** `https://example.com/blog/drupalcons/singapore`

**Route (static):** `/app/blog/drupalcons/singapore/page.js`

**Dynamic route:** `/app/blog/drupalcons/[segment]/page.js`

`/app/blog/[category]/[title]/page.js`

`/app/blog/[...segment]/page.js`





## HTML on Server

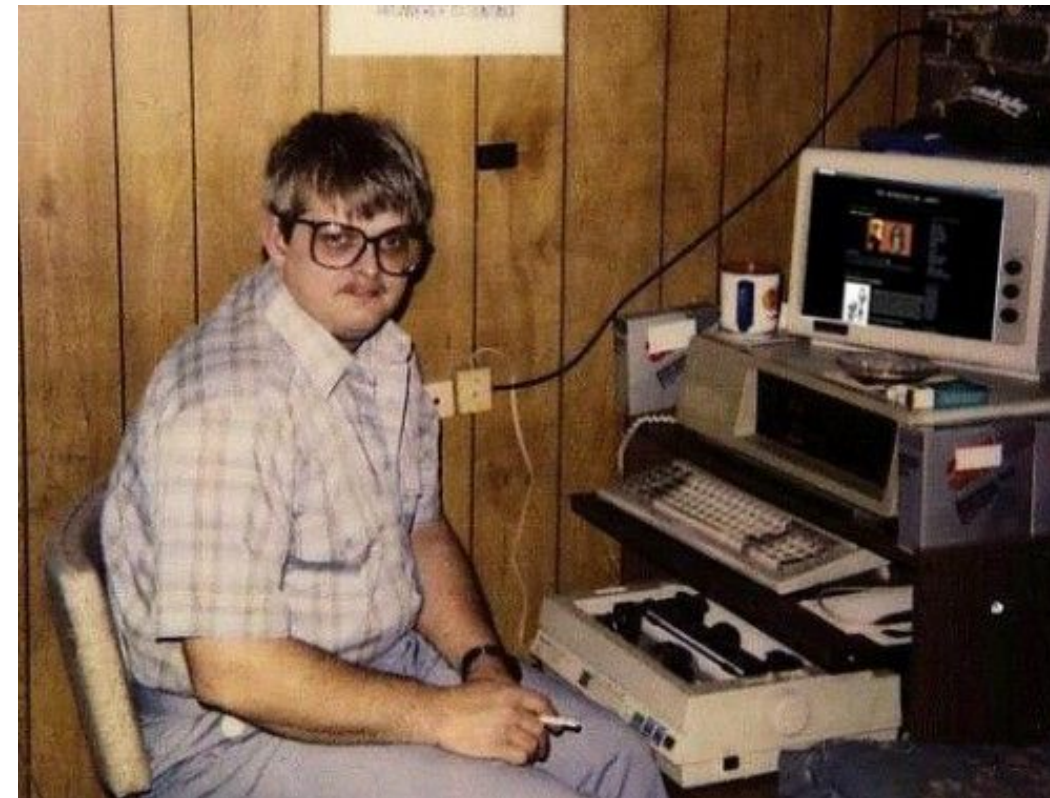
**Drupal**

**React v18  
(client only)**

**React v18  
(with rehydration)**

**React v19 Server  
Components**

## BUILD



dynamic per route

dynamic per path

dynamic per path

## SERVE

static:  
serve big HTML  
w/ small JS per path

dynamic per path

static per route

static:  
HTML per path  
JS per route

static:  
HTML per path  
JS per route

## CLIENT

client JS renders  
*minimal* HTML

client JS renders  
*minimal* HTML

client JS renders  
*all* HTML

client JS re-renders  
*all* HTML

client JS renders  
*minimal* HTML



# Migrating from Pages to App Router

---

Can be done **per route**  
Instead of **all-at-once**





# Next.js App Router

---

```

  app
  [...slug]
    page.js 2024/12
  admin
    page.js 2024/12
  settings
    page.js 202
  blog
    [slug]
      page.js 202
      specs
        page.js
      page.js 2024/12
  components
    Paragraph
      index.js 202
    layout.js 2024/9/13
    page.js 2024/12/8,
```



# Next.js App Router

layout.js

template.js

error.js

loading.js

not-found.js

page.js



Component Hierarchy

```
<Layout>
  <Template>
    <ErrorBoundary fallback={<Error />}>
      <Suspense fallback={<Loading />}>
        <ErrorBoundary fallback={<NotFound />}>
          <Page />
        </ErrorBoundary>
      </Suspense>
    </ErrorBoundary>
  </Template>
</Layout>
```





# Defining Dynamic Routes on Server

```
// File: app/blog/drupalcons/[segment]/page.js
```

```
export async function generateStaticParams() : Promise<[{segment: string}, {segment: ...}> {
```

```
  // Use an async function HERE to retrieve data.
```

```
  return [
```

```
    {
```

```
      segment: 'singapore',
```

```
    },
```

```
    {
```

```
      segment: 'atlanta',
```

```
    },
```

```
    {
```

```
      segment: 'vienna',
```

```
    }
```

```
  ];
```

```
}
```

Use any async function to retrieve data:

- Read data from a file!
- Read data from a database!
- Query JSON:API using Next.js' implementation of Fetch API!
- Use Next Drupal as a helper to retrieve JSON:API from Drupal.



# Defining Dynamic Routes on Server

```
// File: app/blog/[category]/[title]/page.js
```

```
export async function generateStaticParams() : Promise<[{category: string, title: str...}> {  
  return [  
    {  
      category: 'drupalcons',  
      title: 'singapore',  
    },  
    {  
      category: 'drupalcons',  
      title: 'atlanta',  
    },  
    {  
      category: 'music',  
      title: 'blackpink',  
    },  
  ],  
};  
}
```





# Defining Dynamic Routes on Server

// File: app/blog/[...slug]/page.js

```
export async function generateStaticParams() : Promise<[{slug: string[]}, {slug: stri...}> {  
  return [  
    {  
      slug: ['drupalcons', 'singapore'],  
    },  
    {  
      slug: ['drupalcons', 'atlanta'],  
    },  
    {  
      slug: ['that-one-weird-link-bait-about-banana-slugs'],  
    }  
  ];  
}
```



# Rendering Dynamic Routes on Server

*// File: app/blog/drupalcons/[segment]/page.js*

```
export default async function DrupalconPage({
  params: { segment },
  searchParams,
}) : Promise<any> {
  let node
  try {
    node = await getNode(segment)
  } catch (error) {
    // If getNode throws an error, tell Next.js the path is 404.
    notFound()
  }

  return <Article node={node} />;
}
```



# Server components: no client code!

```
import { useEffect, useState } from "react"
import Component from "../Component"
```

Show usages

```
export function OpenNavButtonClient() {
  const [isOpen: boolean, setIsOpen] = useState(initialState: false)
```

```
  useEffect(effect: () => {
    // Do some effect in the web browser.
  }, deps: [])
```

Show usages

```
function buttonHandler(): void {
  // Do something.
  setIsOpen(value: true)
}
```

```
return (
  <Component onClick={buttonHandler} />
)
}
```





# Suspense boundaries and React rehydration

"use client"

```
import { useEffect, useState } from "react"
import Component from "../Component"
```

Show usages

```
export function OpenNavButtonClient() {
  const [isOpen: boolean, setIsOpen] = useState(initialState: false)
```

```
  useEffect( effect: () => {
    // Do some effect in the web browser.
  }, deps: [])
```

Show usages

```
function buttonHandler(): void {
  // Do something.
  setIsOpen( value: true)
}
```

```
return (
  <Component onClick={buttonHandler} />
)
}
```



# Client Components and the "use client" directive

"use client"

```
import { useEffect, useState } from "react"
import Component from "./Component"
```

Show usages

```
export function OpenNavButtonClient() {
  const [isOpen :boolean , setIsOpen] = useState( initialState: false)
```

```
  useEffect( effect: () => {
    // Do some effect in the web browser.
  }, deps: [])
```

Show usages

```
function buttonHandler() :void {
  // Do something.
  setIsOpen( value: true)
}
```

```
return (
  <Component onClick={buttonHandler} />
)
```

```
}
```

OpenNavButton

JS Client.js

JS Component.js

JS index.js



# Client Components and the "use client" directive

OpenNavButton

JS Client.js

JS Component.js

JS index.js



```
export default function OpenNavButton(props) {  
  return (  
    <button className="my-button-style" {...props}>  
      Open  
    </button>  
  )  
}
```





# Client Components and the "use client" directive

```
import { Suspense } from "react";  
import { OpenNavButtonClient } from "../Client";  
import Component from "../Component";
```

▼ OpenNavButton

JS Client.js

JS Component.js

JS index.js



```
export default function OpenNavButton() {  
  return (  
    <Suspense fallback={<Component />}>  
      <OpenNavButtonClient />  
    </Suspense>  
  )  
}
```



# Next Drupal

---

## Drupal modules

<https://www.drupal.org/project/next>

```
composer require drupal/next
```

## NPM package

<https://www.npmjs.com/package/next-drupal>

```
npm install drupal-next@canary
```



# **How the Next module improves the Drupal content editor experience**

---

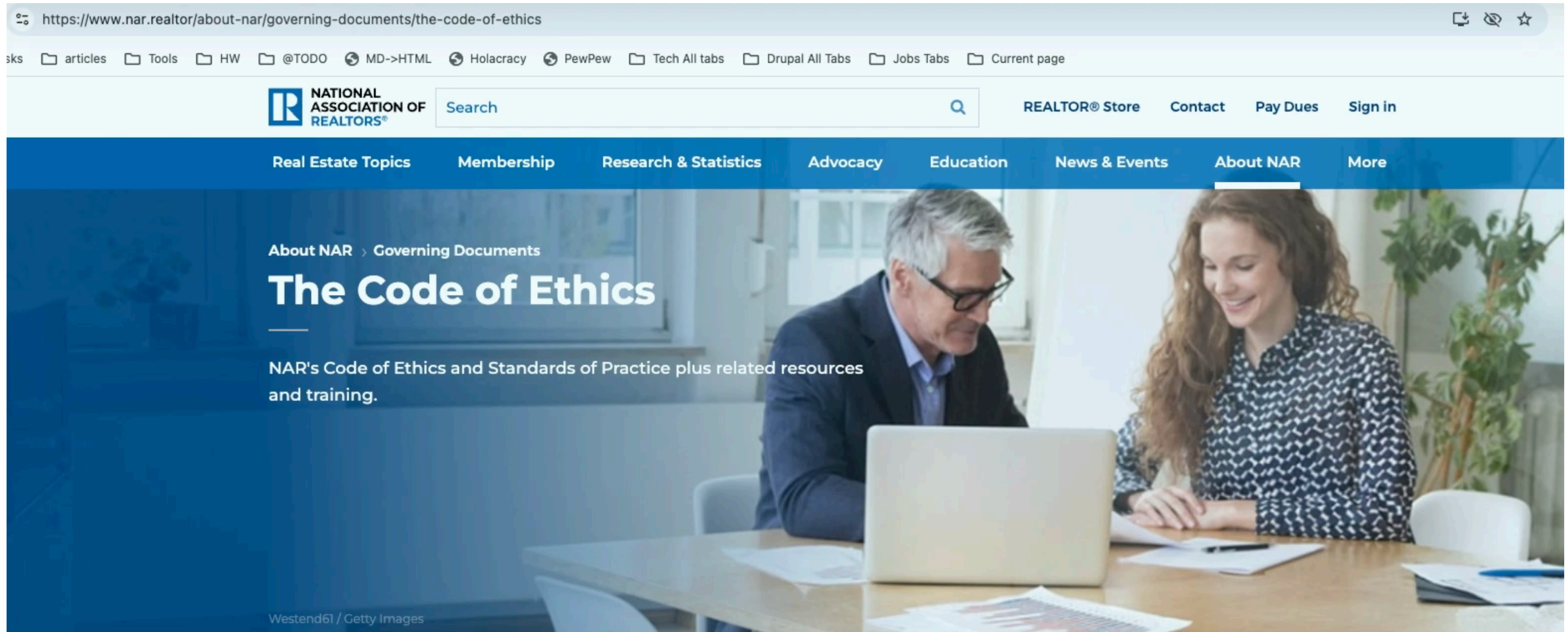
# LIVE DEMO

OH GOD. OH GOD. OH GOD. OH GOD.



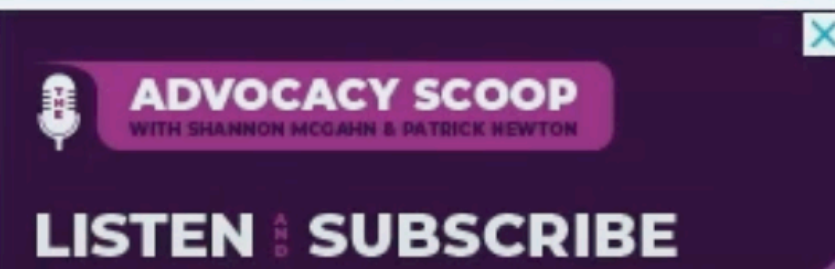


# How the Next module improves the Drupal content editor experience



Share

NAR's Code of Ethics, adopted in 1913, was one of the first codifications of ethical duties adopted by any business group. The Code ensures that consumers are







# How the Next module improves the Drupal content editor experience

The screenshot displays the Drupal content editor interface for the 'The Code of Ethics' page. The top navigation bar includes icons for Content, Structure, Appearance, Extend, People, Reports, NAR, Configuration, and Help. Below this, a breadcrumb trail shows 'Edit The Code of Ethics'. The main title 'The Code of Ethics' is prominently displayed, followed by a set of tabs: View (selected), Edit, Delete, Revisions, Entityqueue, Devel, and JSON:API. A status bar indicates the page is 'Published' with a 'View' button. The content area shows the NAR logo and a search bar. A blue navigation bar lists various sections: Real Estate Topics, Membership, Research & Statistics, Advocacy, Education, News & Events, About NAR, and More. The main content area features a large image of a man and a woman working on a laptop, with the text 'About NAR > Governing Documents' and 'The Code of Ethics' displayed. Below this, a description reads: 'NAR's Code of Ethics and Standards of Practice plus related resources and training.' The footer of the content area credits 'Westend61 / Getty Images'.





# How the Next module improves the Drupal content editor experience

 [Content](#) [Structure](#) [Appearance](#) [Extend](#) [People](#) [Reports](#) [NAR](#) [Configuration](#) [Help](#)

[← Back to site](#) | [Edit Page](#)

## The Code of Ethics ☆

[View](#) [Edit](#) [Delete](#) [Revisions](#) [Entityqueue](#) [Devel](#) [JSON:API](#)

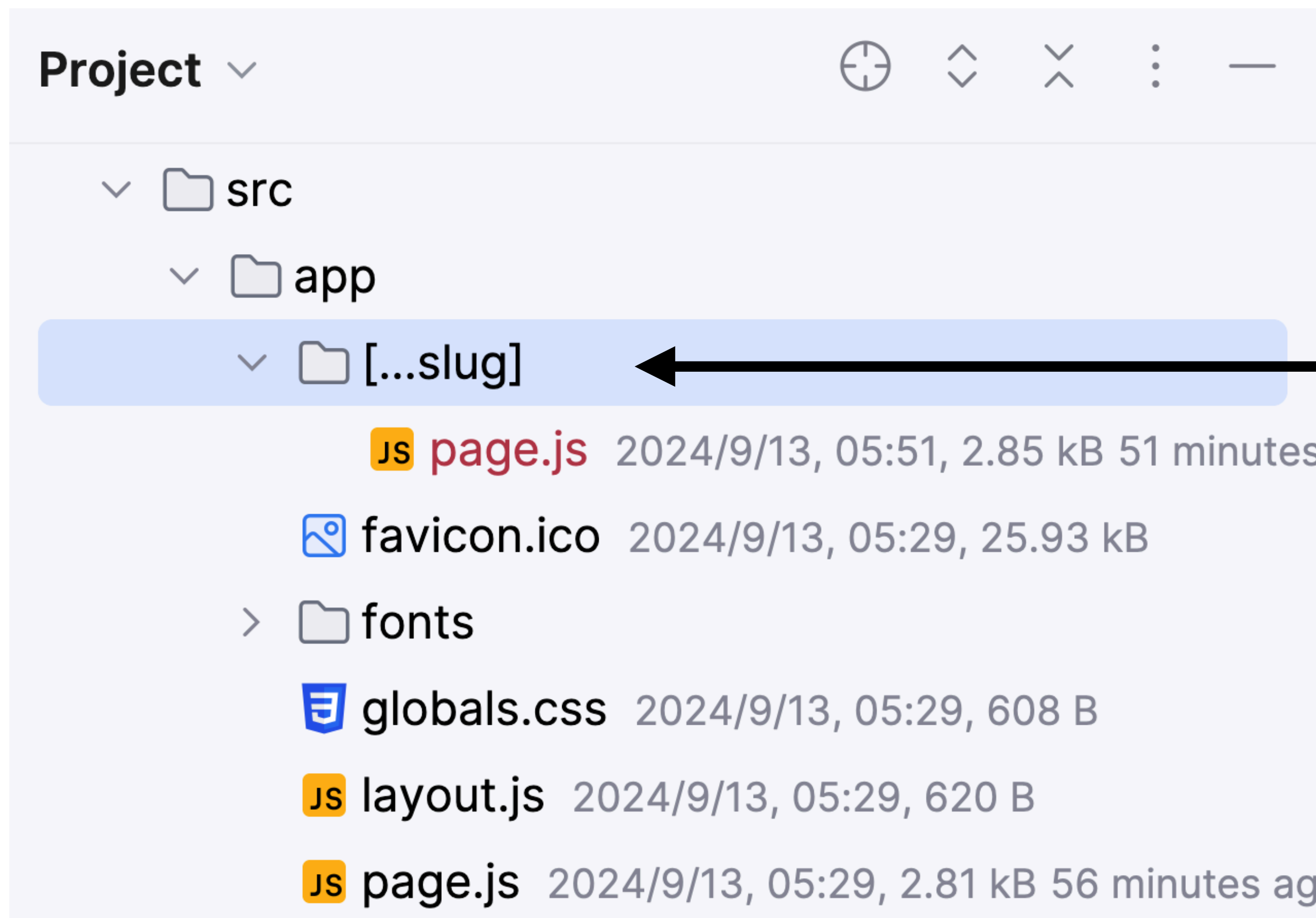
Revisions allow you to track differences between multiple versions of your content, and revert to older versions.

| Revision                                                                          | Operations               |
|-----------------------------------------------------------------------------------|--------------------------|
| 09/13/2024 - 04:50 by admin                                                       | Current revision         |
| 09/13/2024 - 04:48 by admin                                                       | <a href="#">Revert</a> ▼ |
| 09/13/2024 - 04:47 by admin<br>Copy of the revision from Mon, 07/01/2024 - 14:35. | <a href="#">Revert</a> ▼ |
| 09/13/2024 - 04:44 by admin                                                       | <a href="#">Revert</a> ▼ |
| 07/01/2024 - 14:35 by stephaniec3                                                 | <a href="#">Revert</a> ▼ |





# Tip: the perils of a catch-all route



Using one Next.js  
Dynamic Route for  
ALL paths  
defined by Drupal



# Tip: the perils of a catch-all route

PHP imports are *dynamic*

if a *path* needs Paragraphs, import its code

```
require_once DRUPAL_ROOT . '/' . $include;
```

JavaScript imports are *static*

if a *route* needs Paragraphs, import its code

```
import { Paragraph } from './components/Paragraph';
```





# Tip: the perils of a catch-all route

---

**ALL COMPONENTS  
ARE IMPORTED  
FOR ALL ROUTES**



# Tip: the perils of a catch-all route

---

- ★ **Constrain Drupal paths with pathauto;  
one Next.js Dynamic Route *per* pathauto pattern**
- ★ **Constrain Paragraph types  
*per* Field**
- ★ **“Unicorn” Paragraph types (React Components)  
should use static routes**





**Thank you!**





# Questions?