



Orlando

**Code Camp**



Transitioning to the  
next phase of web  
development.







# AGENDA

01

## Welcome

Agenda / Intro

04

## Frameworks

Svelte-Kit, Remix, Astro, NextJS,  
NuxtJS

02

## Current State

Website vs SPA

05

## Themes

Shared themes and features

03

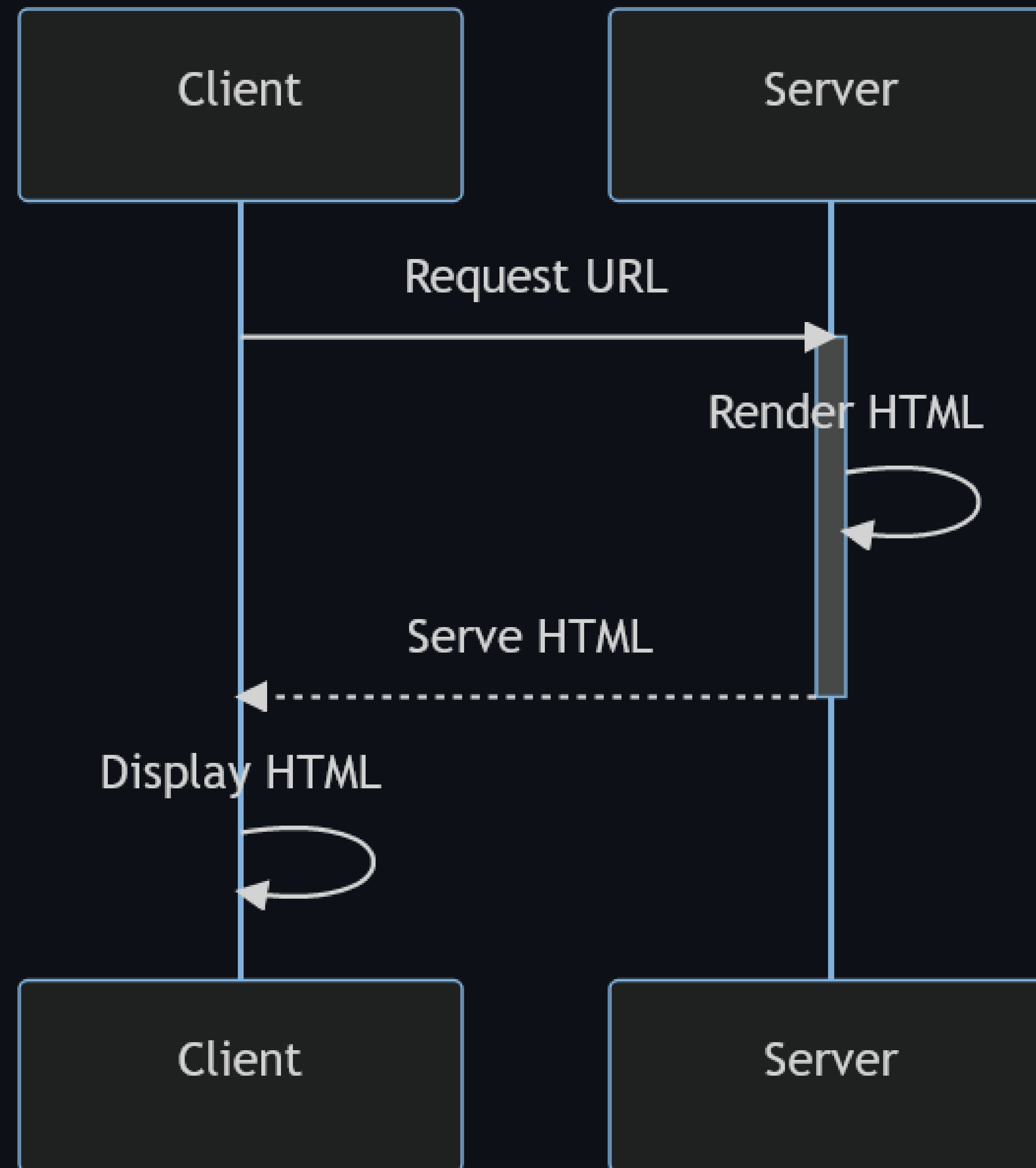
## New Wave

What this new wave of frameworks  
hopes to accomplish.

06

## Thank You

Grab a sticker!



```
<html>
  <head>
    <title>Basic HTML</title>
  </head>
  <body>
    <h1>Basic HTML</h1>
    <p><a href="https://www.domain.com/page"></p>
    <form method="get">
      <input type="text" name="name" />
      <input type="submit" value="Submit" />
    </form>
    <form method="post">
      <input type="text" name="name" />
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>
```



+ For you Following

Staff Picks

Anna Burgess Yang

Egor Romanov · Jan 20

Building a Startup from Scratch: CTO

When I was first approached to help build the startup, I had yet to learn what I was getting in friend to audit the solution that the previous te

Startup 13 min read · Selected for you

IcePanel · Jan 24

Top 9 software architecture tools

The best free and paid tools to visualize your s ⚡ TL;dr Software architecture tools can be cat groups, modelling tools, diagrams as code and

Software Architecture 6 min read · Selected for you

José Carlos · Nov 3, 2022 ⭐ Member-only

How I made ~5\$ per day — in Pas: an android app)

by José Paiva TL;DR: In this article I talk about android phone to make around 5\$ per day (wi

SIGN IN

SUBSCRIBE



TODAY'S PICKS



Silicon Valley Will Still Need A Bank

GIDEON LICHFIELD



TV

After *The Last of Us*, Everything Will Be Transmedia

WILL BEDINGFIELD



DOMINO RU

THE S CONT

The collaps is about to

CHRIS STOK

MOST RECENT

TikTok and Meta's Moderators



WIKIPEDIA The Free Encyclopedia

English

6 629 000+ articles

Русский

1 900 000+ статей

Español

1 846 000+ artículos

Deutsch

2 781 000+ Artikel

Italiano

1 801 000+ voci

Português

1 102 000+ artigos

日本語

1 366 000+ 記事

Français

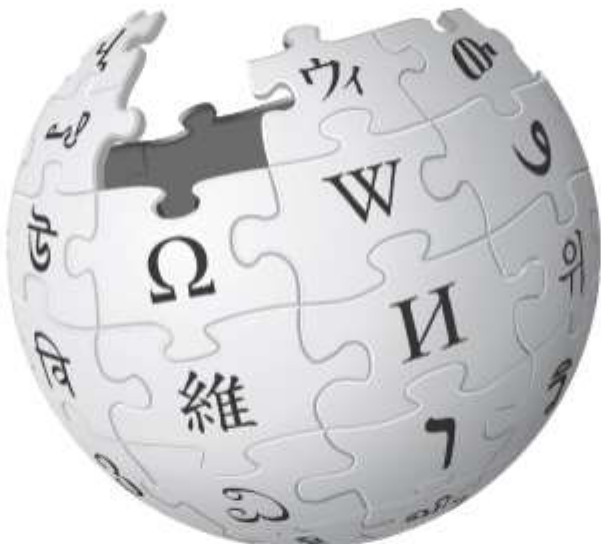
2 504 000+ articles

中文

1 340 000+ 条目 / 條目

العربية

1 202 000+ مقالة



EN



Read Wikipedia in your language



Wikipedia is hosted by the Wikimedia Foundation, a non-profit organization that also hosts a range of other projects. You can support our work with a donation.



Download Wikipedia for Android or iOS

Save your favorite articles to read offline, sync your reading lists across devices and customize your reading experience with the official Wikipedia app.



Commons  
Freely usable photos & more



Wikibooks  
Free textbooks



Wikiversity  
Free course materials



Wikivoyage  
Free travel guide



Wikinews  
Free news source



Wikiquote  
Free quote



Wiktionary  
Free dictionary



Wikidata  
Free knowledge base



MediaWiki  
Free & open wiki



mootools

 **jQuery**  
*write less, do more.*

dōjō  
toolkit

*script.aculo.us*  
*it's about the user interface, baby!*

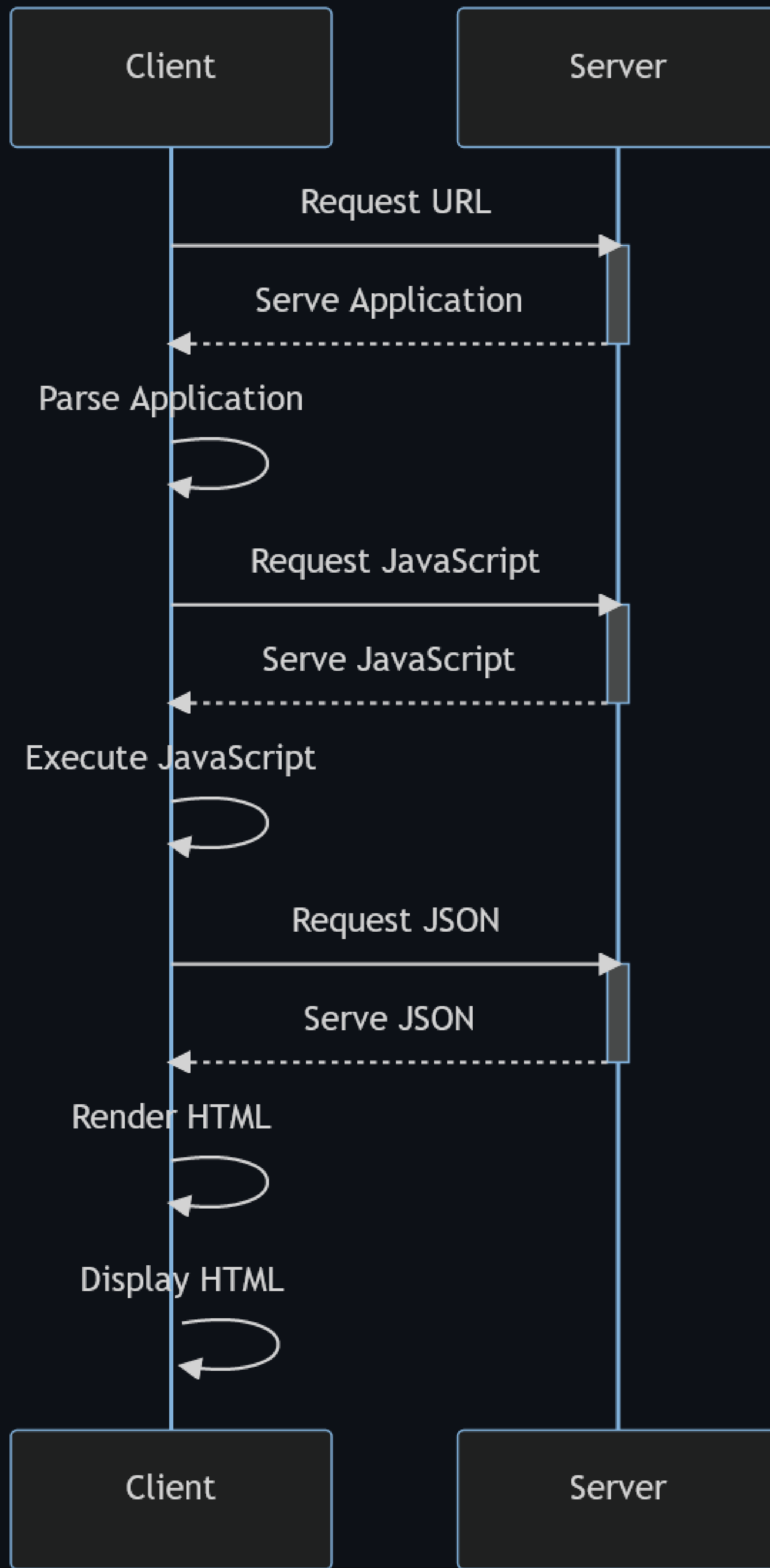


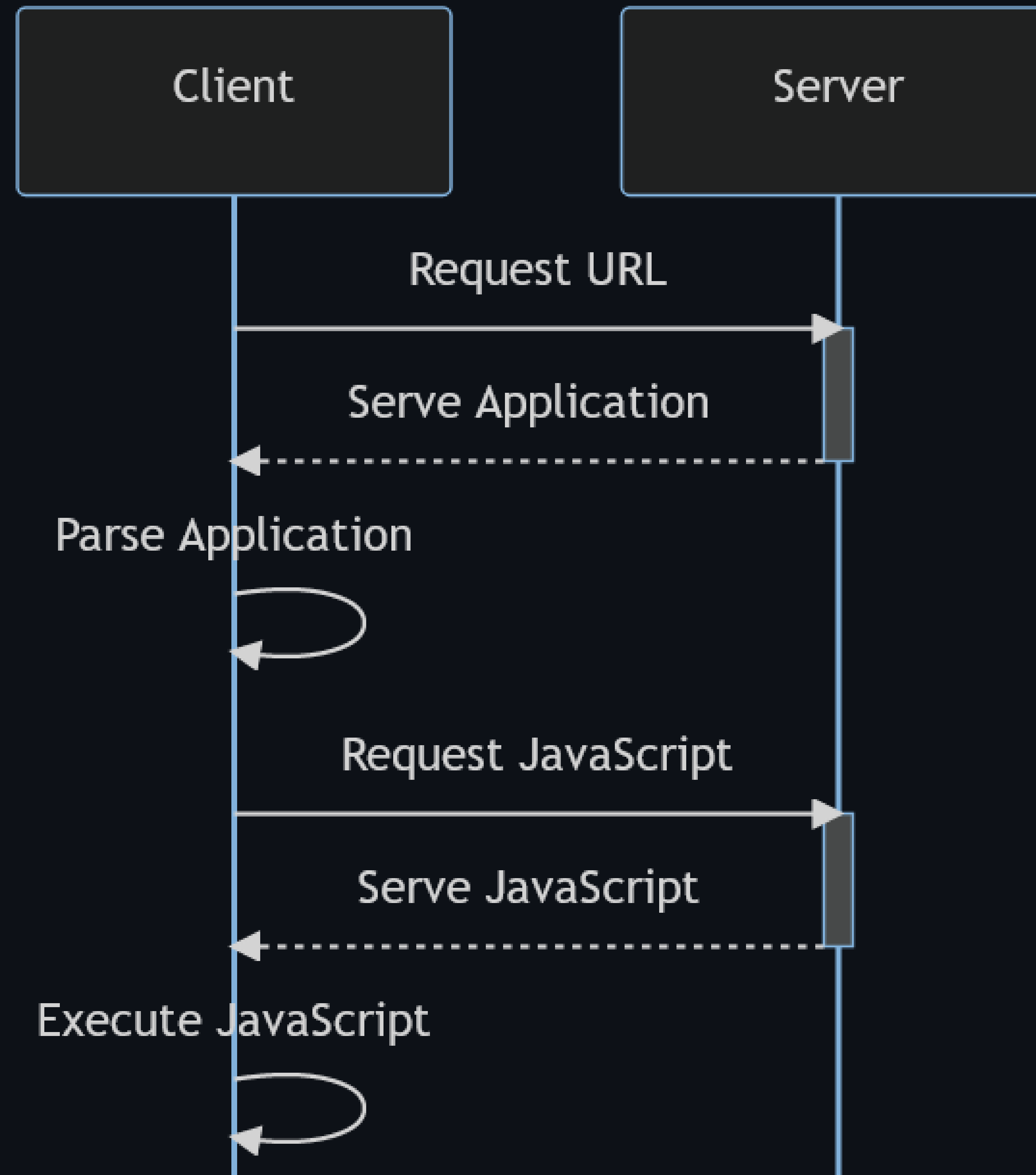
**Alpine.js**

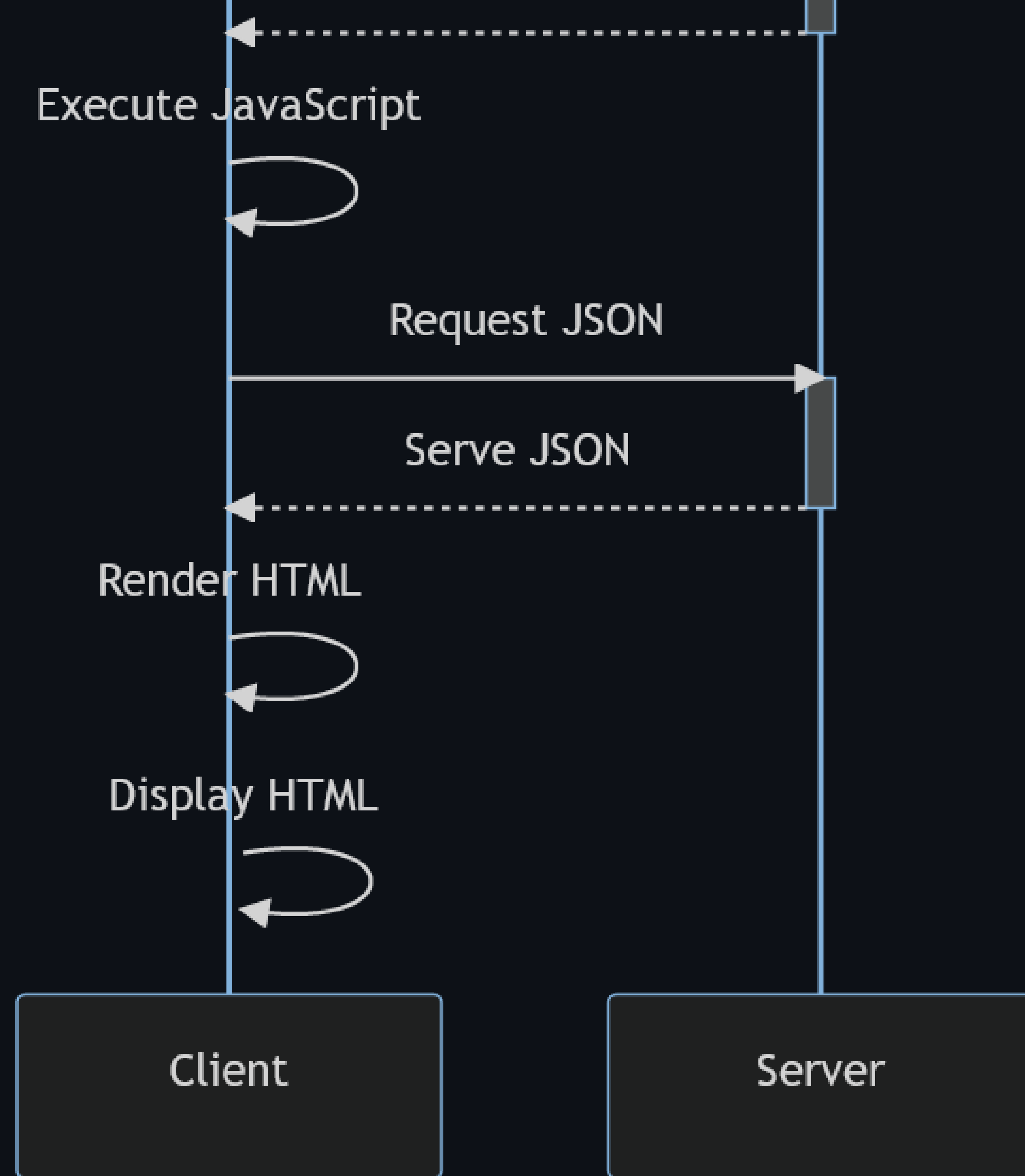
# Single Page Application













Orlando, FL

Orlando

Florida

Mostly sunny · 76°F  
4:27 PM

Directions

Save

Nearby

Send to phone

Share

Orlando, a city in central Florida, is home to more than a dozen theme parks. Chief among its claims to fame is Walt Disney World, comprised of parks ... [More](#)

Iconic Orlando

Walt Disney World® Resort

4.7 ★ (236,709)

Universal Orlando Resort

4.7 ★ (162,252)

Magic Kingdom Park

4.6 ★ (198,314)

Disney's Animal Kingdom Theme Park

4.7 ★ (113,365)

Epcot

4.7 ★ (126,290)

Restaurants

Hotels

Things to do

Museums

Transit

Layers

in

Search

Home

My Network

Jobs

Messaging

Notifications

Me

Tools

Advertise

James Luterek

Tech Guide | Speaker | Developer Advocate @ Elastic Path

Who's viewed your profile

149

Impressions of your posts

536

Access exclusive tools & insights

Reactivate Premium: 10% Off

My items

RivalHound

Page notifications  
0

Page visitors  
0

See visitor analytics

Recent

No Code API Orchestration | ...

Kibo Commerce Quarterly Up...

E-Commerce Network

AngularJS, React JS, Java Full S...

AWS Certified Solutions Archi...

Groups

E-Commerce Network

Start a post

Photo

Video

Audio event

Write article

Tracy Lee (She/Her) · 1st

CEO at This Dot Labs, Staff Augmentation | Goo...

5h · 🌐

What is your favorite quote or personal mantra that keeps you inspired?

2

Like

Comment

Repost

Send

Snowflake

577,409 followers

Promoted

Join us for a free instructor-led, hands-on lab to learn how easy it is to turn your organization into a ...see more

LinkedIn News

Regulators battle SVB fallout

Top news · 16,256 readers

Food giant plans \$5 million daycare

20m ago · 1,372 readers

More high earners keep renting

Search Triangle Devs

# javascript /3w

3,641

so you could somehow run in a hot refresh mo... Today when you wanted or let it use a production-matching version at other times, but never finished that

spence-s

11:07 AM

yeah thank you for the input here - it sounds like there is a similar problem here you are solving for as well, but haven't found the exact solution. I'm still too new to know all the little details of how it all works currently, but luckily there are some other wins available here to make it more palatable. Some people use <https://www.npmjs.com/package/relative-deps> in development a lot to help their workflows, but its still insane to have to wait between 1-2min between changes for some workflows.

relative-deps

Installs local dependencies for optimal developer experience. Latest version: 1.0.7, last published: 2 years ago. Start using relative-deps in your project by running `npm i relative-deps`. There are no other projects in the npm registry using relative-deps.

I whipped up a custom gulp script that sits outside of all the component repos, watches them, rebuilds them, and manually installs them in the host repo, so at the very least, it would be totally automated after a change, but this feels very very hacky.

Nathan Fries

11:41 AM

I have been using Vite for module federation, it has been a much more pleasant experience than Webpack - but ofc much of that comes down to simply being so much faster

Eli Weinstock-Herman

12:16 PM

for us having the http proxy (and also proxying the websocket for HMR/react refresh/whatever) solves this, as we don't have to wait for a full rebuild either since it's a second active dev server instance rebuilding on the fly

I've gone down the npm link route before for shared component libraries, but I don't know if I got refresh working (I know I spent a lot of time trying, can't remember if I succeeded somehow)

👍 1

B I S | | | | | | | | | |

Message #javascript

+

javascript

15



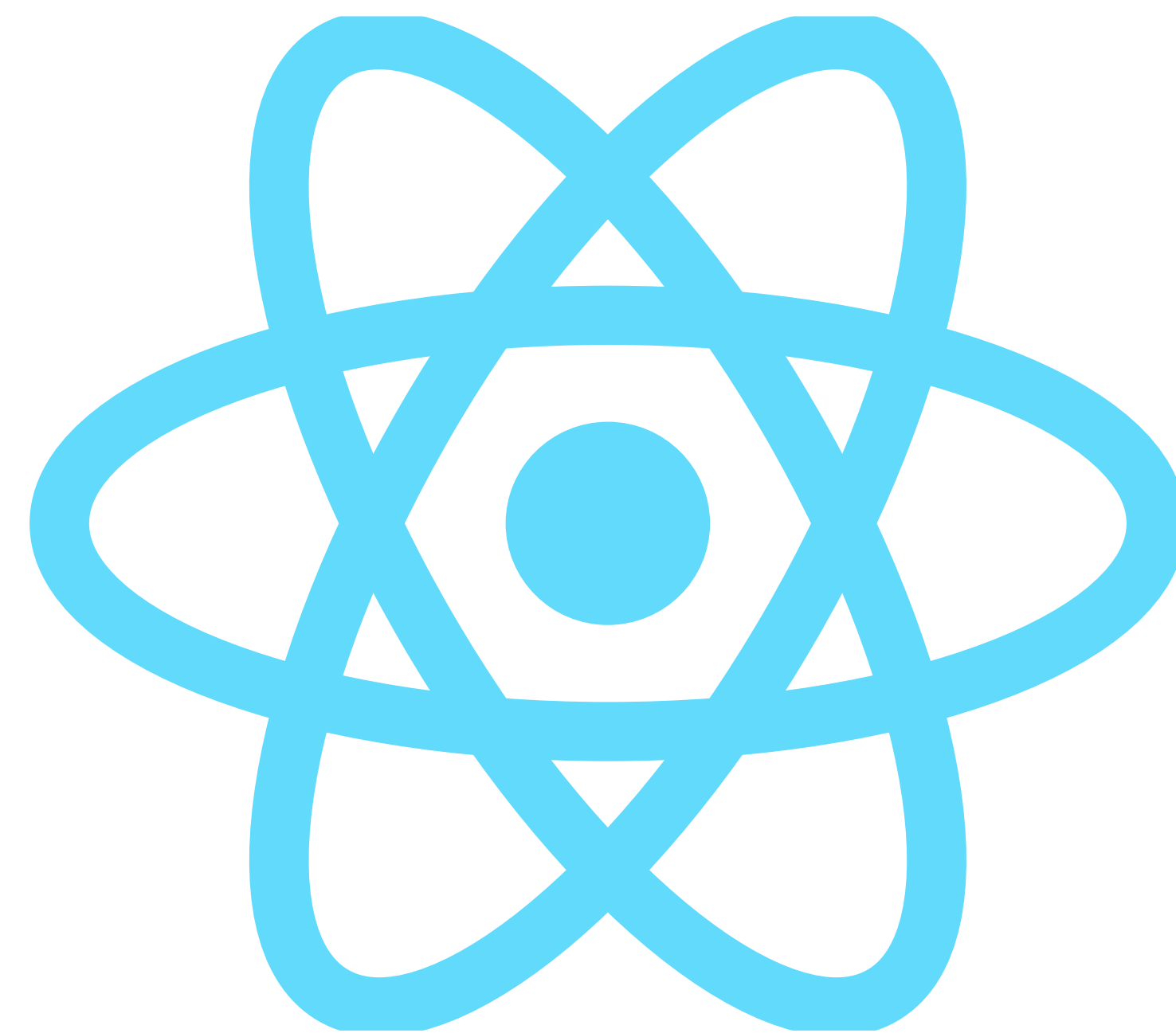
*Knockout.*



BACKBONE.JS

*Knockout.*

 BACKBONE.JS





```
// Requesting data from server
fetch('https://api.example.com/data.json')
  .then(response => response.json())
  .then(data => {
    console.log(data);
  })
  .catch(error => {
    console.error(error);
  });

// Setting the page / URL for the browser
window.history.pushState(null, null, "/new-url");
```

# SPA vs PWA

Single Page Application  
Progressive Web App

# A LIST APART

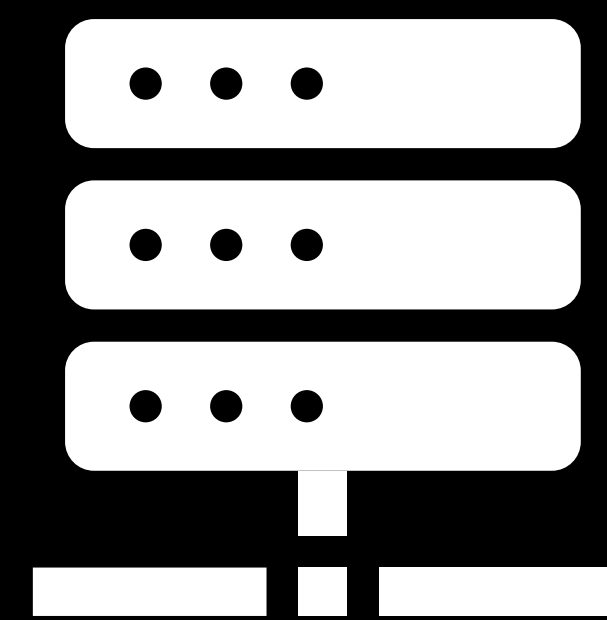
Understanding Progressive Enhancement

<https://alistapart.com/article/progressiveenhancementwithjavascript/>

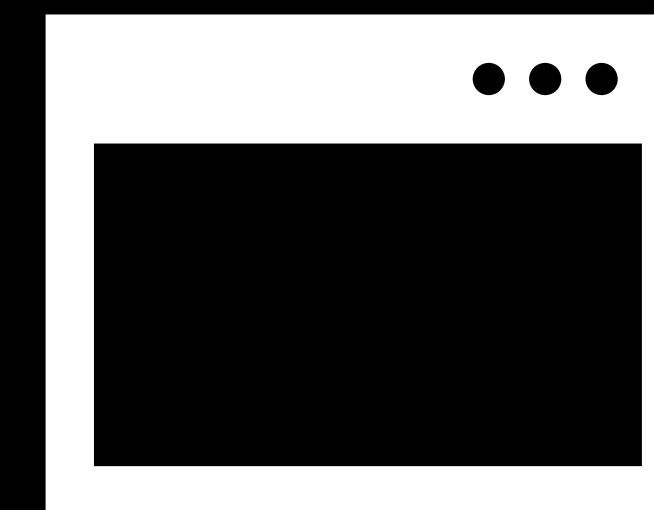


# Next Wave





Server  
Rendered

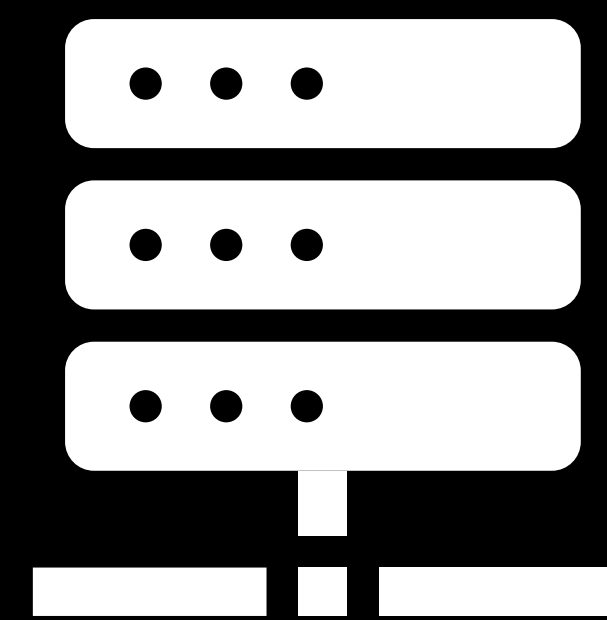


SPA / PWA

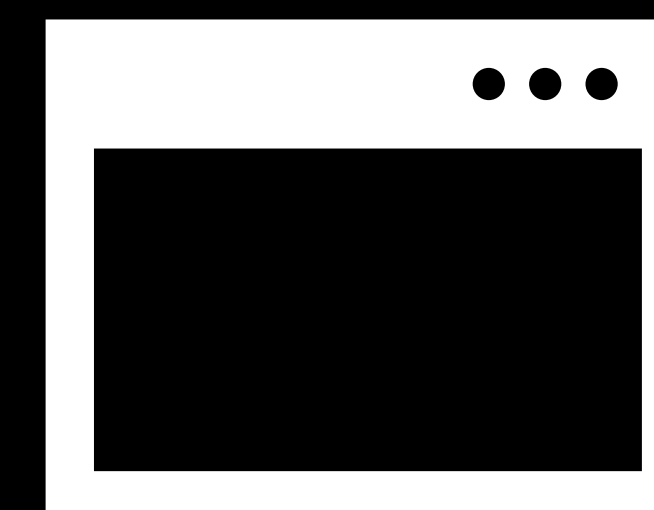




**WHY NOT BOTH?**

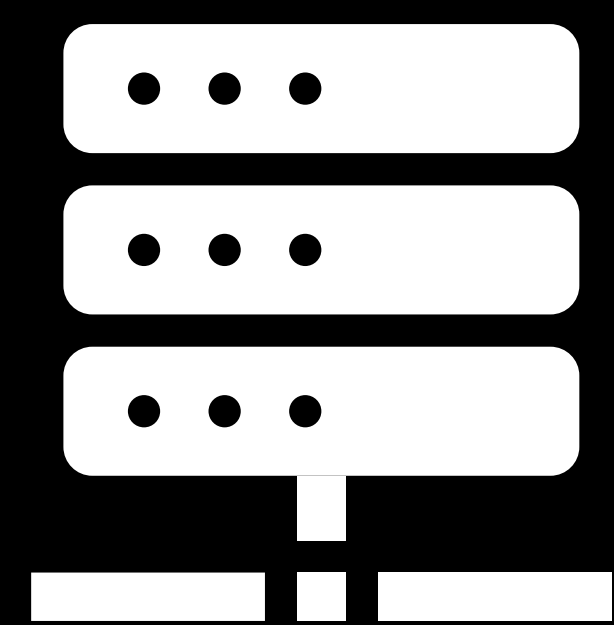


Server  
Rendered



SPA / PWA





Server  
Rendered

Static SSR  
(JAMStack)

Partial  
Hydration

Prerending  
& Hydration

SPA with  
Prerendering



SPA / PWA



MPA (Multi-Page Application)

Hybrid Frameworks

# Transitional Apps



**Remix**



~~NEXT~~.JS





SVELTEKIT





- Compiled
- Full-Stack
- Server-side APIs
- Supports Edge Computing
- Server-rendering



# SVELTEKIT

## File System Routing

✓ routes

⌘ +layout.svelte

⌘ +page.svelte

TS +page.ts

⌘ Counter.svelte

⌘ Header.svelte

# styles.css

> about

> sverdle

✓ test

⌘ +error.svelte

⌘ +layout.svelte

⌘ +loading.svelte

⌘ +page.svelte

TS +page.ts

TS +server.ts

routes/  
+page  
+error  
+layout  
+server

# SVELTEKIT +page.svelte Rendering

```
<script lang="ts">
```

```
</script>
```

# SVELTEKIT +page.svelte Rendering

```
<script lang="ts">
```

```
  // Build time render (static / JAMStack)  
  export const prerender = true;
```

```
</script>
```

# SVELTEKIT +page.svelte Rendering

```
<script lang="ts">  
  
  // Build time render (static / JAMStack)  
  export const prerender = true;  
  // Server-side Rendering (SSR)  
  export const ssr = false;  
  
</script>
```



# SVELTEKIT +page.svelte Rendering

```
<script lang="ts">  
  
  // Build time render (static / JAMStack)  
  export const prerender = true;  
  // Server-side Rendering (SSR)  
  export const ssr = false;  
  // Client-side Rendering (CSR)  
  export const csr = false;  
  
</script>
```



+page.svelte

```
<script lang="ts">

  import type { PageData }
    from './$types';
  export let data: PageData;

</script>

<h1>{data.post.title}</h1>
<div>{@html data.post.content}</div>
```

+page.ts

```
import type { PageLoad } from './$types';

export function load() {
  return {
    post: {
      title: `Title for goes here`,
      content: `Content for goes here`
    }
  };
}
```



## +page.svelte

```
<script lang="ts">

    import type { PageData }
      from './$types';
    export let data: PageData;

</script>

<h1>{data.post.title}</h1>
<div>{@html data.post.content}</div>
```

## +page.server.ts

```
import type { PageLoad } from './$types';

export function load() {
    return {
        post: {
            title: `Title for goes here`,
            content: `Content for goes here`
        }
    };
}
```



## VSCoDe Tips

 *+page.svelte about* X

src > routes > about >  +page.svelte >  svelte:head

```
1 <svelte:head>
```

```
2   <title>About</title>
```

## Label Format

Settings > Label Format > Short

## VSCode Tips

```
TS tsconfig.json
TS vite.config.ts
> .svelte-kit
> node_modules
✓ src
  TS app.d.ts
  <> app.html
  > lib
  ✓ routes
    ⚡ +layout.svelte
    ⚡ +page.svelte
    TS +page.ts
```

### Sort Order Settings

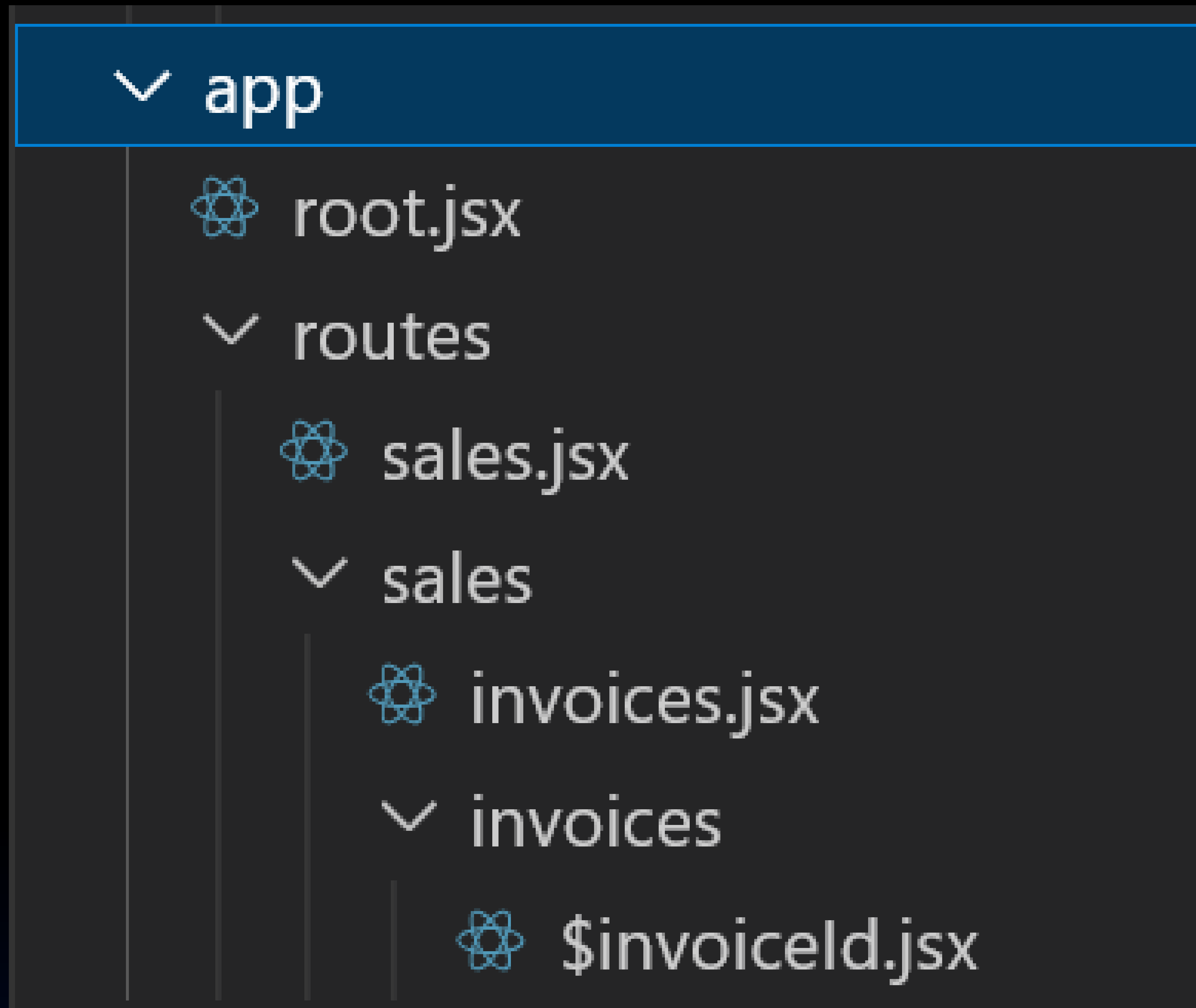
- > Explorer: Sort Order
- > filesFirst

Remix





- Full-Stack ReactJS Framework
- Only supports SSR
- Nested Routes
- Actions for Forms



`<Outlet />`



# Nested Routes

▼ app	
🌀 root.jsx	
▼ routes	
🌀 sales.jsx	
▼ sales	
🌀 invoices.jsx	
▼ invoices	
🌀 \$invoiceId.jsx	

example.com/sales/invoices/102000

**Fakebooks**

Dashboard

Accounts

Sales

Expenses

Reports

## Sales

OverviewSubscriptionsInvoicesCustomersDeposits

OVERDUE  
\$10,800

DUE SOON  
\$62,000

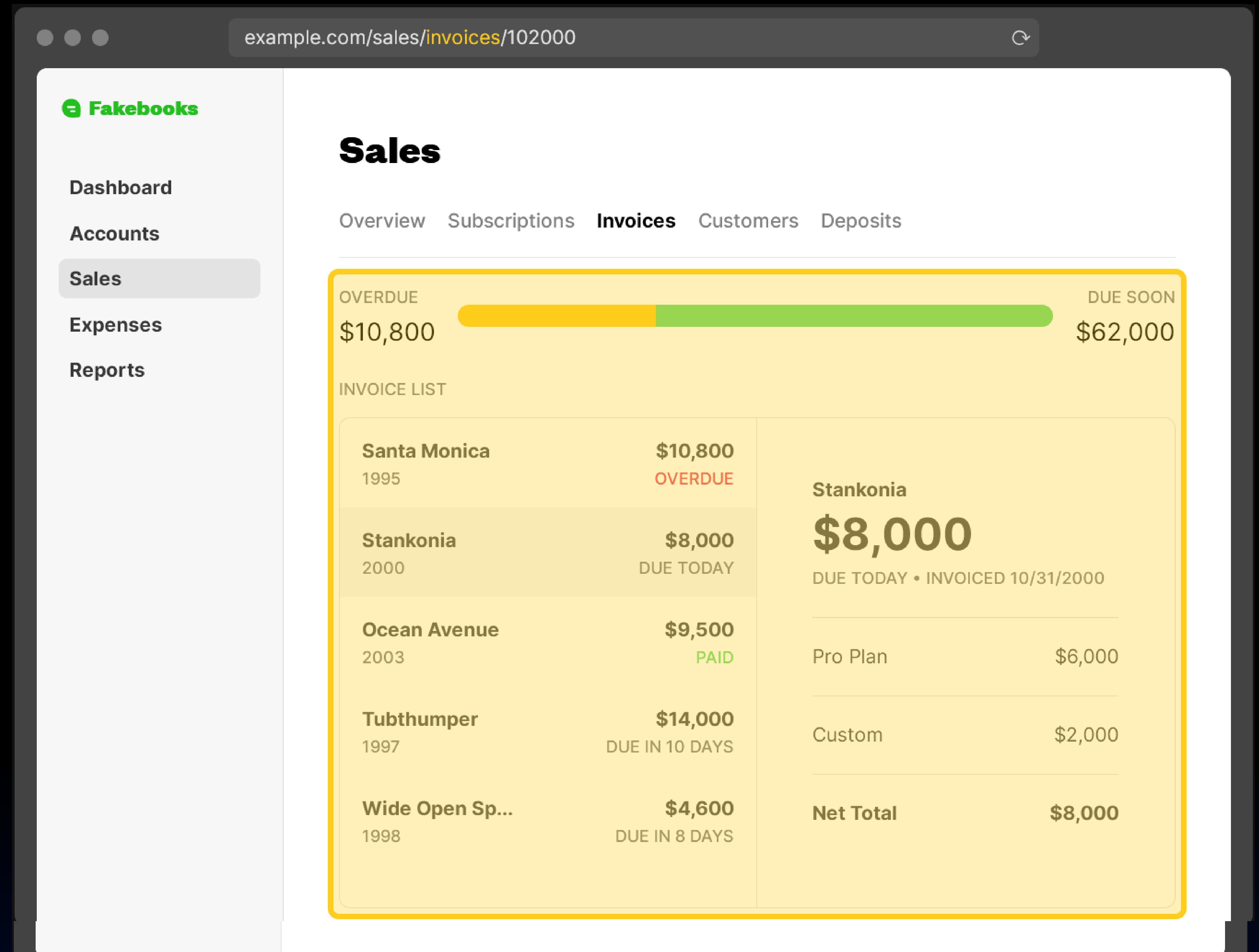
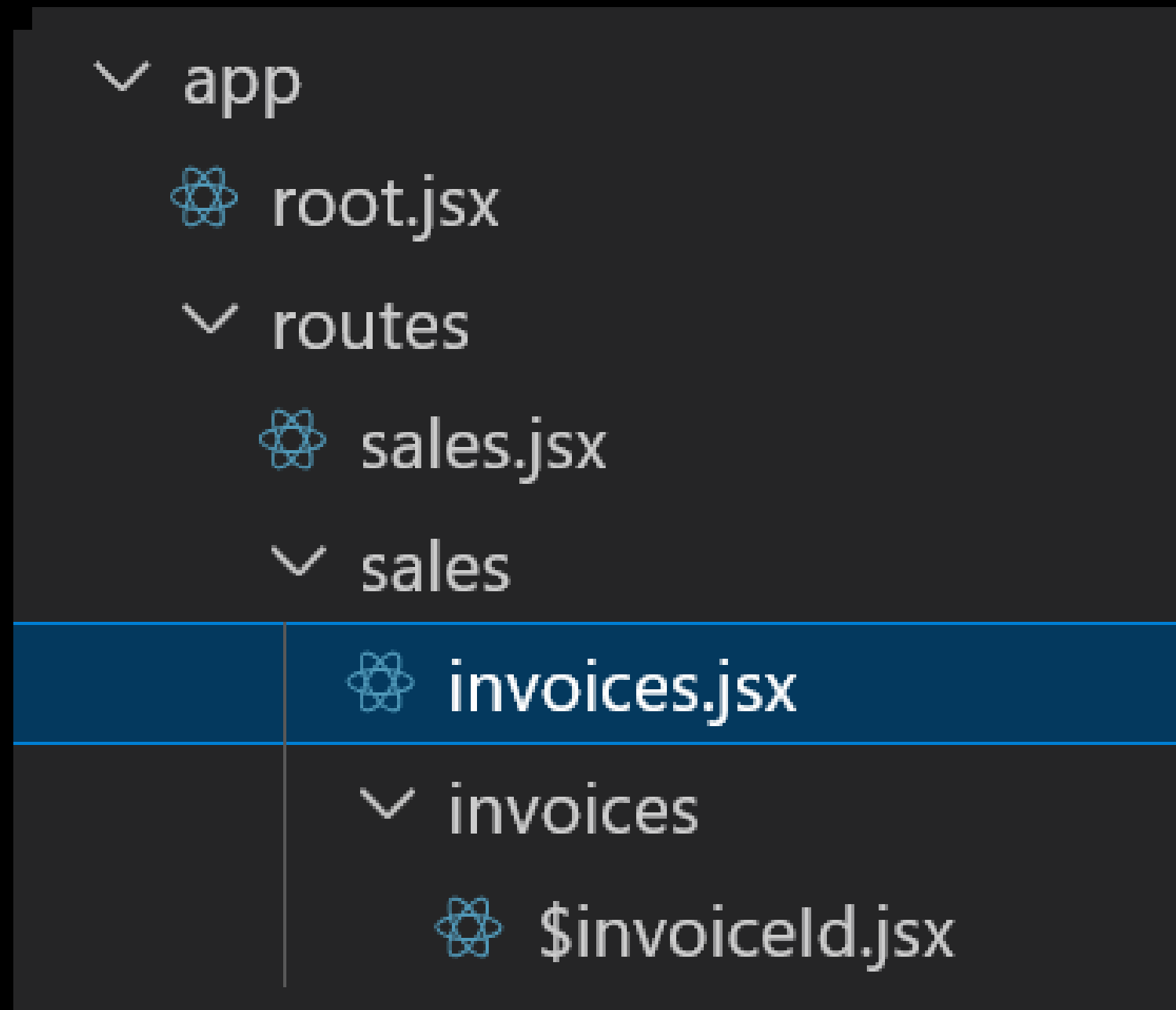
INVOICE LIST

<b>Santa Monica</b> 1995	<b>\$10,800</b> OVERDUE	<div><b>Stankonia</b></div> <div><b>\$8,000</b></div> <div>DUE TODAY • INVOICED 10/31/2000</div> <div><div>Pro Plan\$6,000</div><div>Custom\$2,000</div><div>Net Total\$8,000</div></div>
<b>Stankonia</b> 2000	<b>\$8,000</b> DUE TODAY	
<b>Ocean Avenue</b> 2003	<b>\$9,500</b> PAID	
<b>Tubthumper</b> 1997	<b>\$14,000</b> DUE IN 10 DAYS	
<b>Wide Open Sp...</b> 1998	<b>\$4,600</b> DUE IN 8 DAYS	





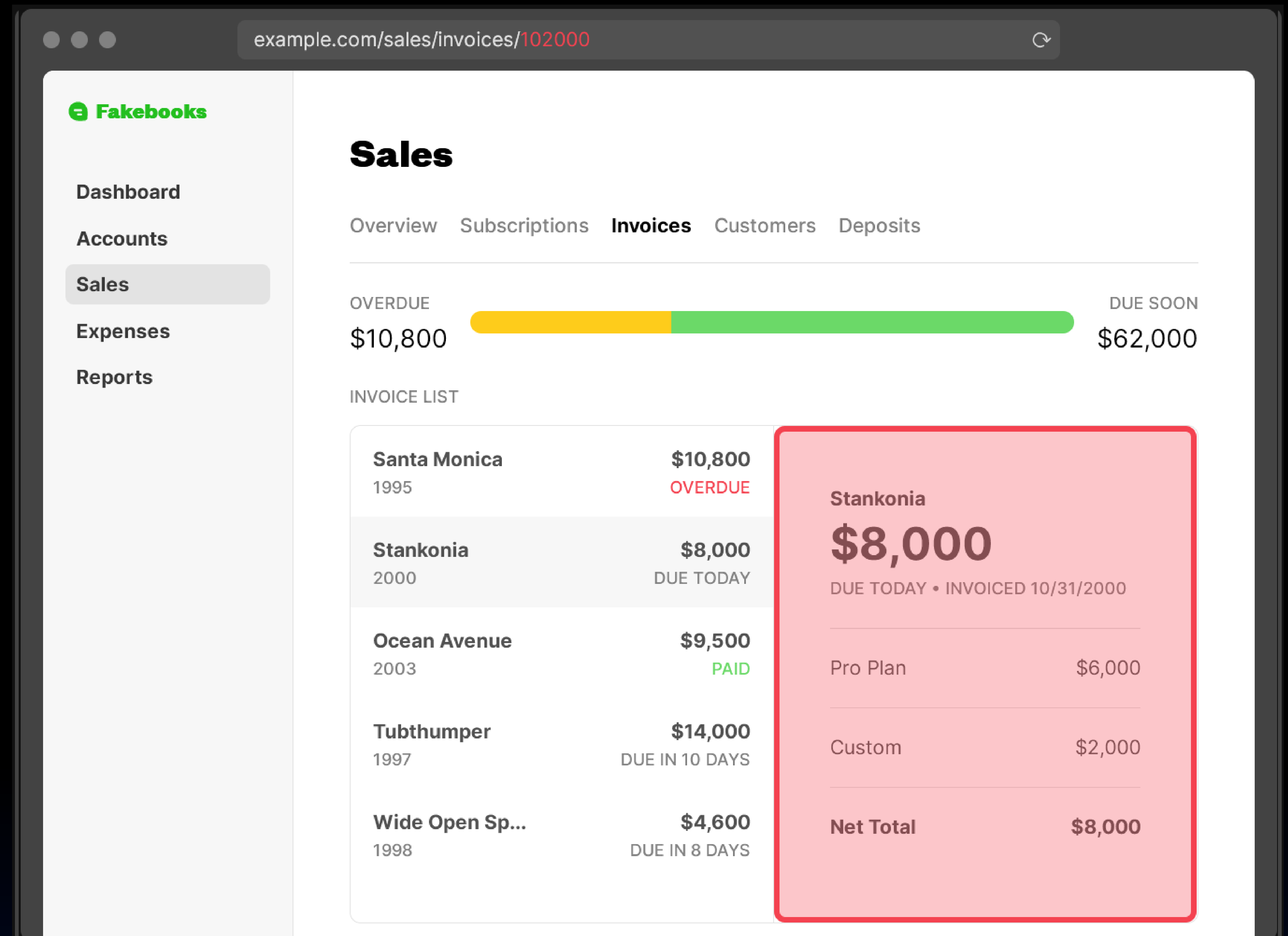
# Nested Routes





# Nested Routes

```
✓ app
  └─ root.jsx
    └─ routes
      └─ sales.jsx
        └─ sales
          └─ invoices.jsx
            └─ invoices
              └─ $invoiceId.jsx
```



```
function OldForm() {  
  handleSubmit(event) {  
    // Prevent normal form behavior  
    event.preventDefault();  
    // do something with the form data  
  }  
  return (  
    <form onSubmit={handleSubmit}>  
      <input type="submit" value="submit" />  
    </form>  
  )  
}
```





## Forms (Remix Way)

```
import { Form } from "@remix-run/react";
export async function action({ request }: ActionArgs) {
  const formData = await request.formData();
  const email = formData.get("email");
}
export default function FormPage() {
  return (
    <Form method="post">
      <input type="email" name="email" />
      <input type="submit" value="submit" />
    </Form>
  )
}
```



# Stacks

The Blues Stack: Deployed to the edge (distributed) with a long-running Node.js server and PostgreSQL database. Intended for large and fast production-grade applications serving millions of users.

The Indie Stack: Deployed to a long-running Node.js server with a persistent SQLite database. This stack is great for websites with dynamic data that you control (blogs, marketing, content sites). It's also a perfect, low-complexity bootstrap for MVPs, prototypes, and proof-of-concepts that can later be updated to the Blues stack easily.

The Grunge Stack: Deployed to a serverless function running Node.js with DynamoDB for persistence. Intended for folks who want to deploy a production-grade application on AWS infrastructure serving millions of users.





- Designed to be server-first
- Works with existing UI frameworks
- No client JS by default
- Supports SSR and SSG
- Selective/Partial hydration



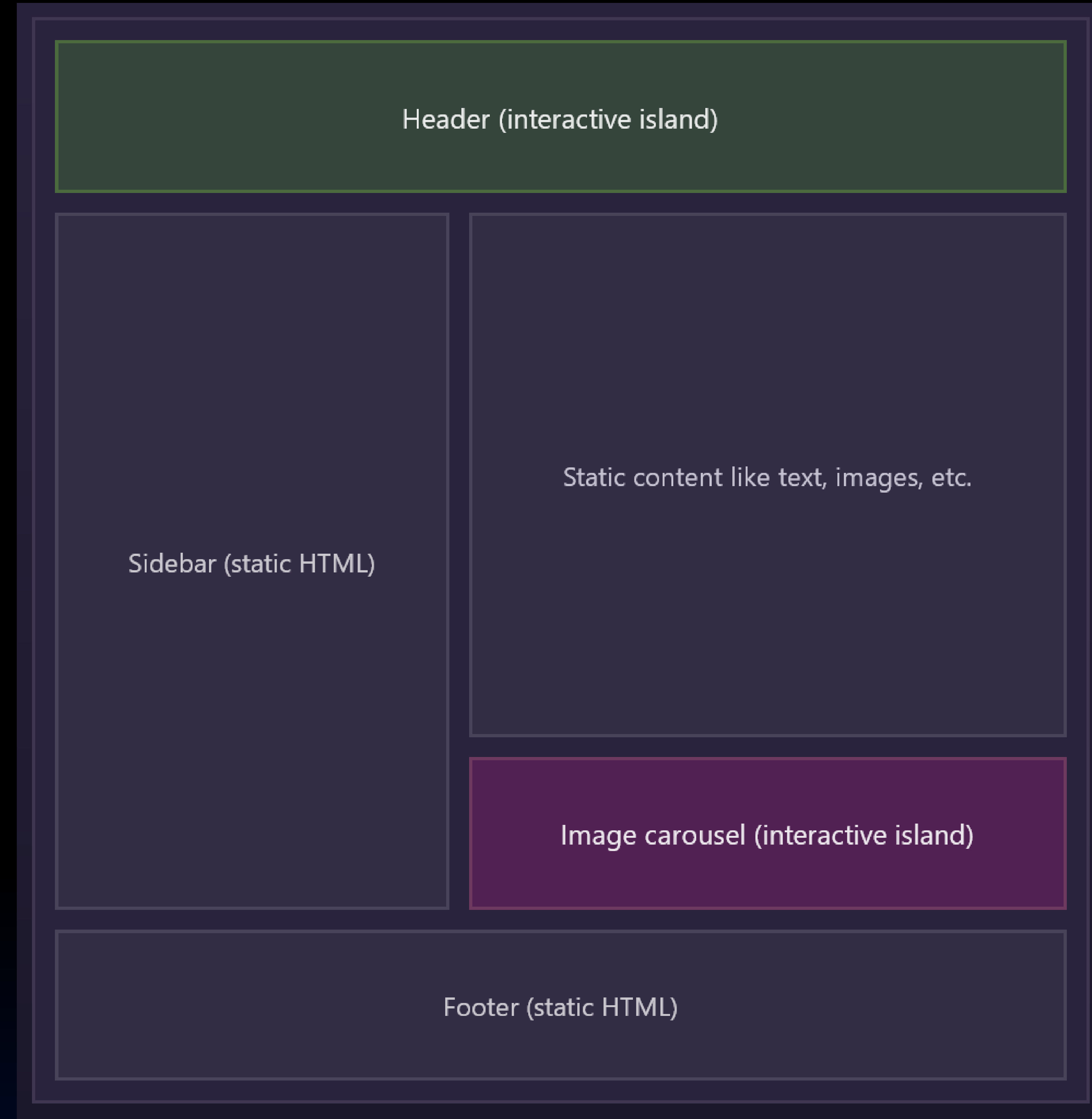




# Astro Islands

```
import MyComponent from  
"./components/MyComponent.jsx";
```

```
<!-- No JavaScript loaded on the page! -->  
<MyReactComponent />
```



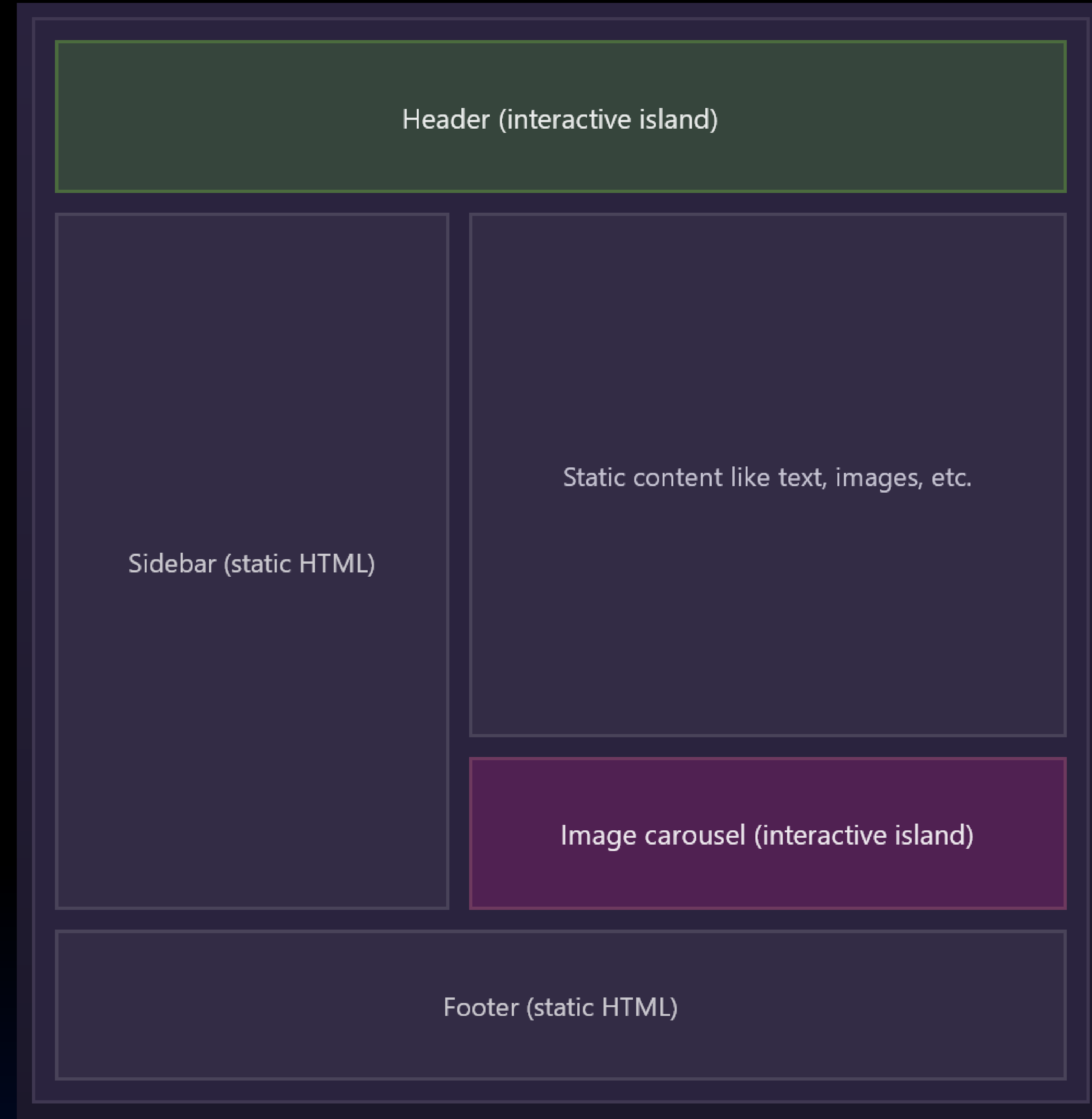


# Astro Islands

```
import MyComponent from  
"./components/MyComponent.jsx";
```

```
<!-- No JavaScript loaded on the page! -->  
<MyReactComponent />
```

```
<!-- Pulled on load -->  
<MyReactComponent client:load />
```





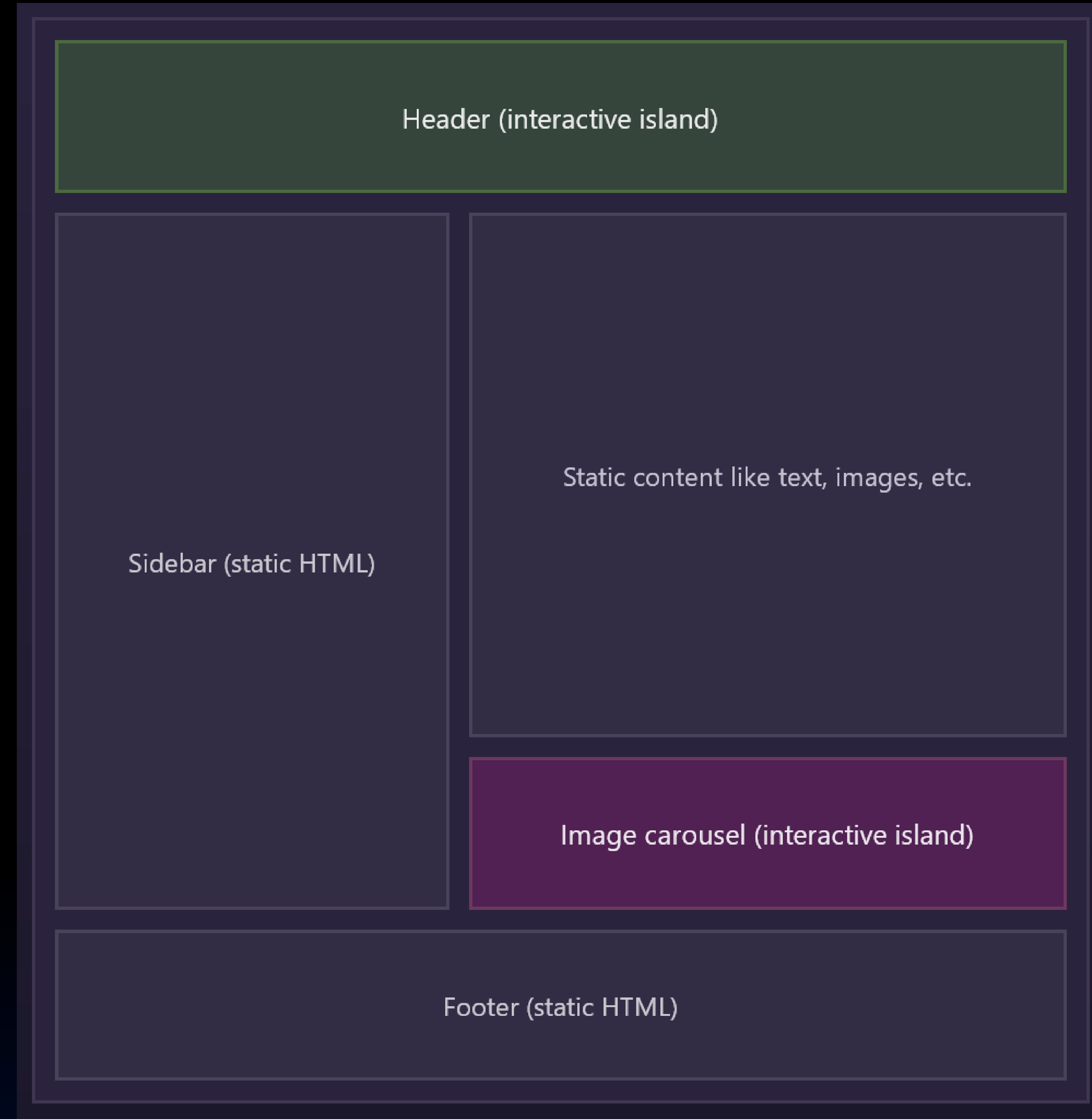
# Astro Islands

```
import MyComponent from
"./components/MyComponent.jsx";

<!-- No JavaScript loaded on the page! -->
<MyReactComponent />

<!-- Pulled on load -->
<MyReactComponent client:load />

<!-- Pulled when browser has time -->
<MyReactComponent client:idle />
```





# Astro Islands

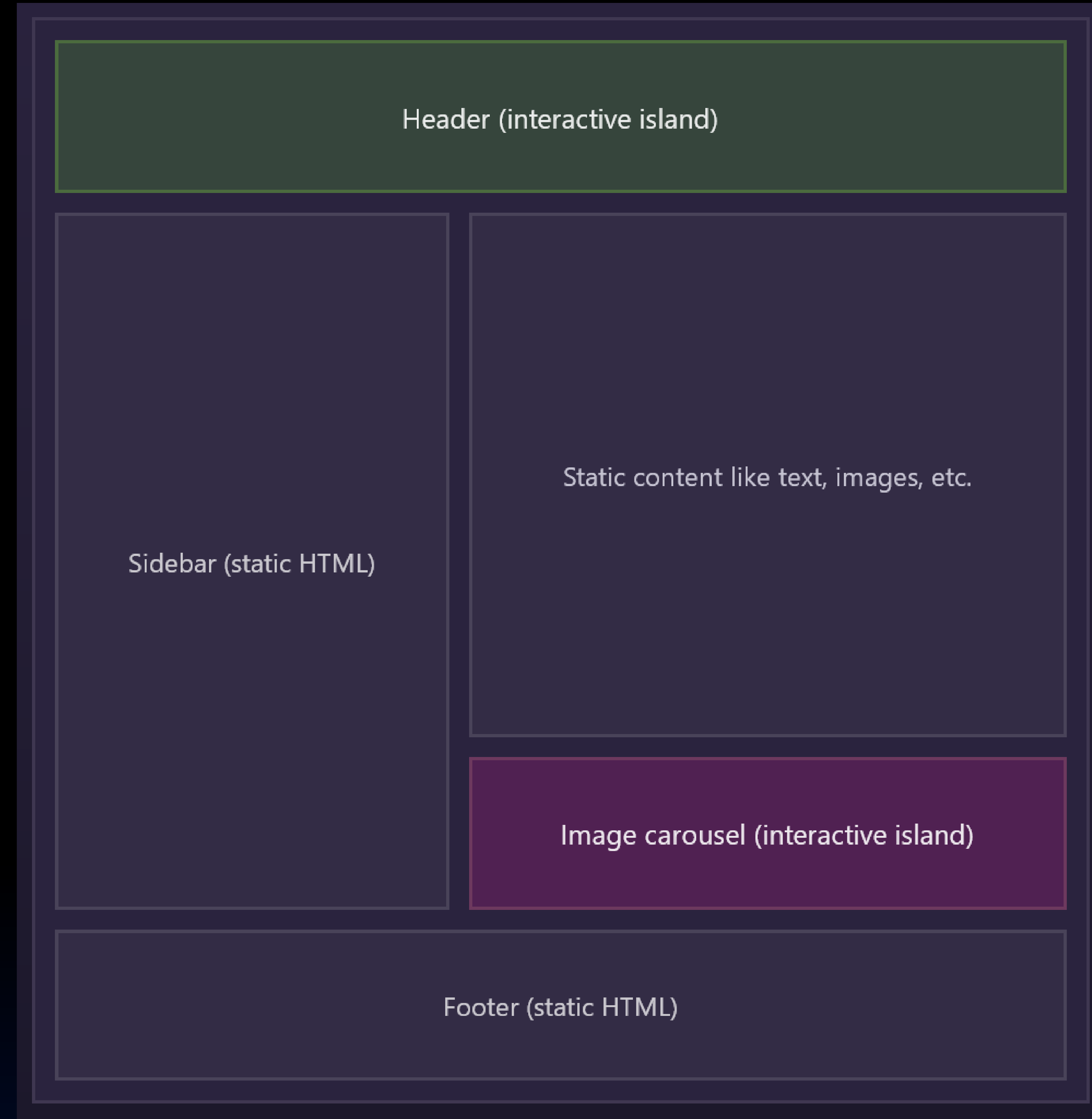
```
import MyComponent from
"./components/MyComponent.jsx";

<!-- No JavaScript loaded on the page! -->
<MyReactComponent />

<!-- Pulled on load -->
<MyReactComponent client:load />

<!-- Pulled when browser has time -->
<MyReactComponent client:idle />

<!-- Pulled when visible -->
<MyReactComponent client:visible />
```







## Endpoints

```
import { getProduct } from '../db';

export async function get({ params }) {
  const id = params.id;
  const product = await getProduct(id);
  if (!product) {
    return new Response(null, {
      status: 404,
      statusText: 'Not found'
    });
  }

  return new Response(JSON.stringify(product), {
    status: 200,
    headers: {
      "Content-Type": "application/json"
    }
  });
}
```



NUXTJS



1. Universal rendering
2. Hybrid rendering
3. Full static mode
4. File-based routing

next.js



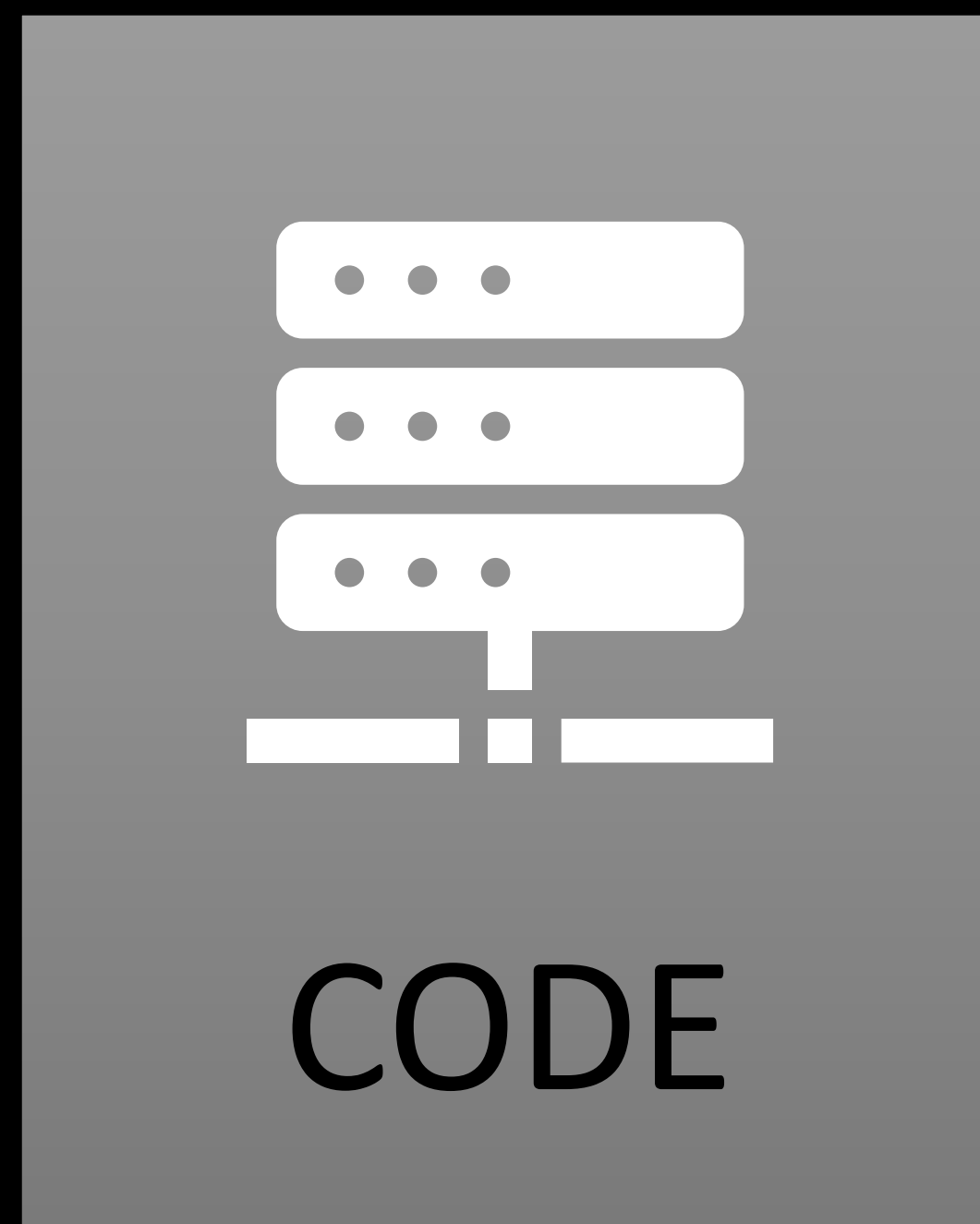


1. Page-based Routing
2. Static file serving
3. Server-side API routes
4. Hybrid Pages
5. React Server Components
6. Better Data Loading

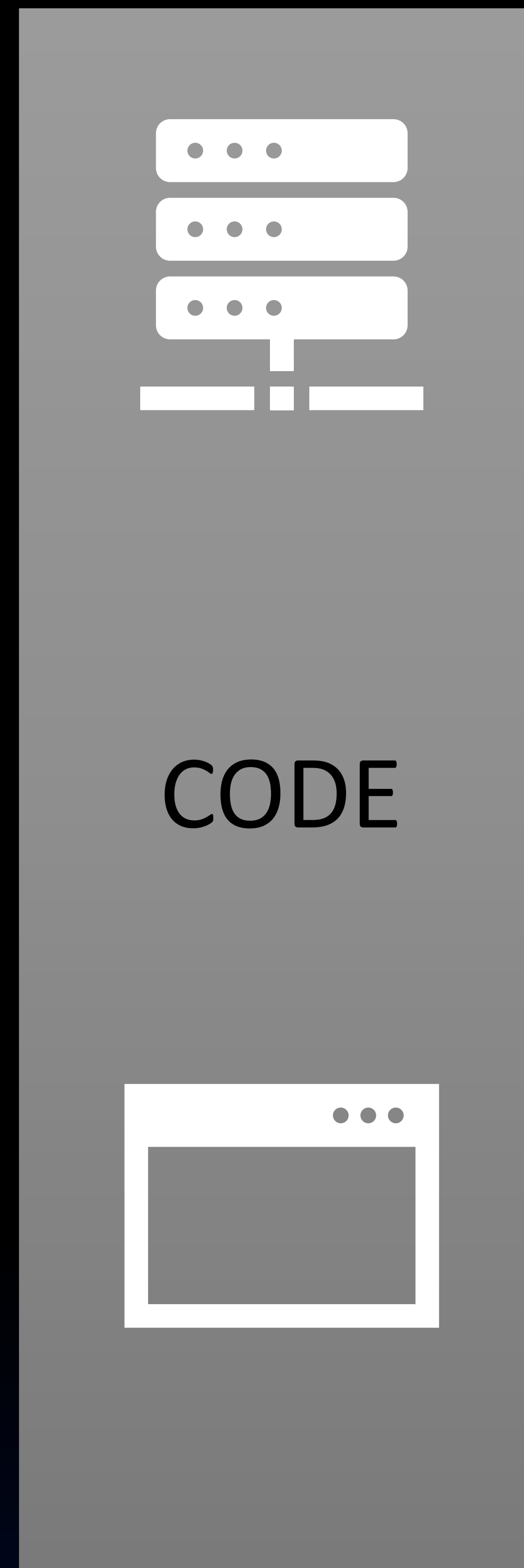
# Common Themes



# Websites



# Transitional Apps



# SPA/PWA



- Full-Stack
- Edge Computing
- Web Standards
- Hybrid Rendering
- Optimized Bundling
- SEO Friendly
- File based routing
- API Functions / Data Loading
- Image Optimization



# Thank You!

