Move over, Gatsby

React Static in Practice

Hello 👋

I'm Nikas

Tech Lead @ Hostmaker

nikas.praninskas.com

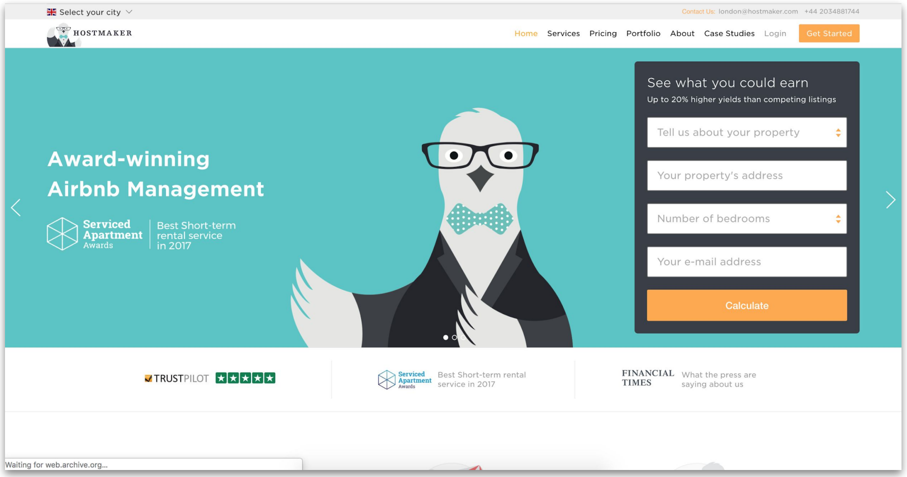
github.com/nikaspran

@nikaspran

Hostmaker.com - 6 months ago

- Slow-ish
- Legacy stack
  - Slow builds (20+ min)
  - Difficult deployment
- Needed a brand refresh

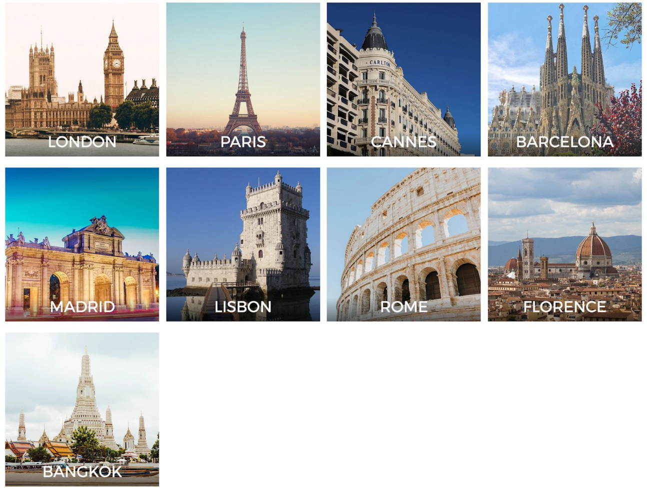
It was time for a rewrite


Select your city
Contact Us: london@hostmaker.com +44 2034881744
HOSTMAKER
Home
Services
Pricing
Portfolio
About
Case Studies
Login
Get Started
Award-winning
Airbnb Management
Serviced Apartment Awards
Best Short-term rental service in 2017
See what you could earn
Up to 20% higher yields than competing listings
Tell us about your property
Your property's address
Number of bedrooms
Your e-mail address
Calculate
TRUSTPILOT
Serviced Apartment Awards
Best Short-term rental service in 2017
FINANCIAL TIMES
What the press are saying about us
Waiting for web.archive.org...

Challenges (1)

- Tons of unique content
  - 9 cities + 6 country hubs
  - 2+ languages each
  - 15+ "templates"
- Dynamically generated content
  - PhraseApp
  - Cosmic JS
  - Greenhouse
  - Our own API
  - ...

1500+ unique pages



Challenges (2)

- Search Engine Optimisation
- Page Speed
  - Don't want to be fetching all that dynamic content on every load
- Avoid the big switch-over
  - Tight deadlines
  - Difficult sell

We were looking for something...

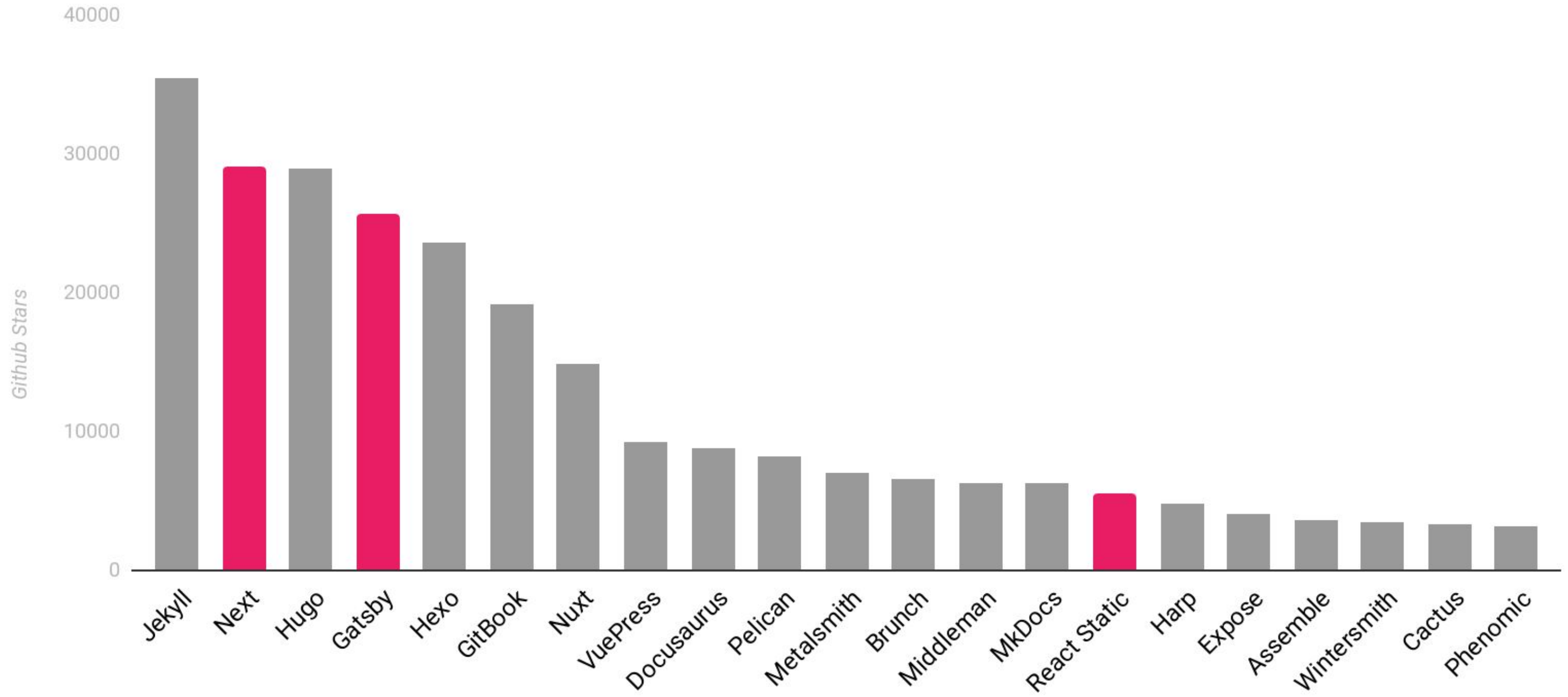
- ⬚ React based
  - Just like all of our other products
- ⬚ Flexible
  - Had to integrate with our existing codebase
  - Had to support iterative migration
- ⬚ Simple
  - We were finishing our migration to React
  - Did not want new tools (i.e. GraphQL) to be a barrier
- ⬚ Long term
  - Simplicity === we can change fetching strategies, add caching, etc.

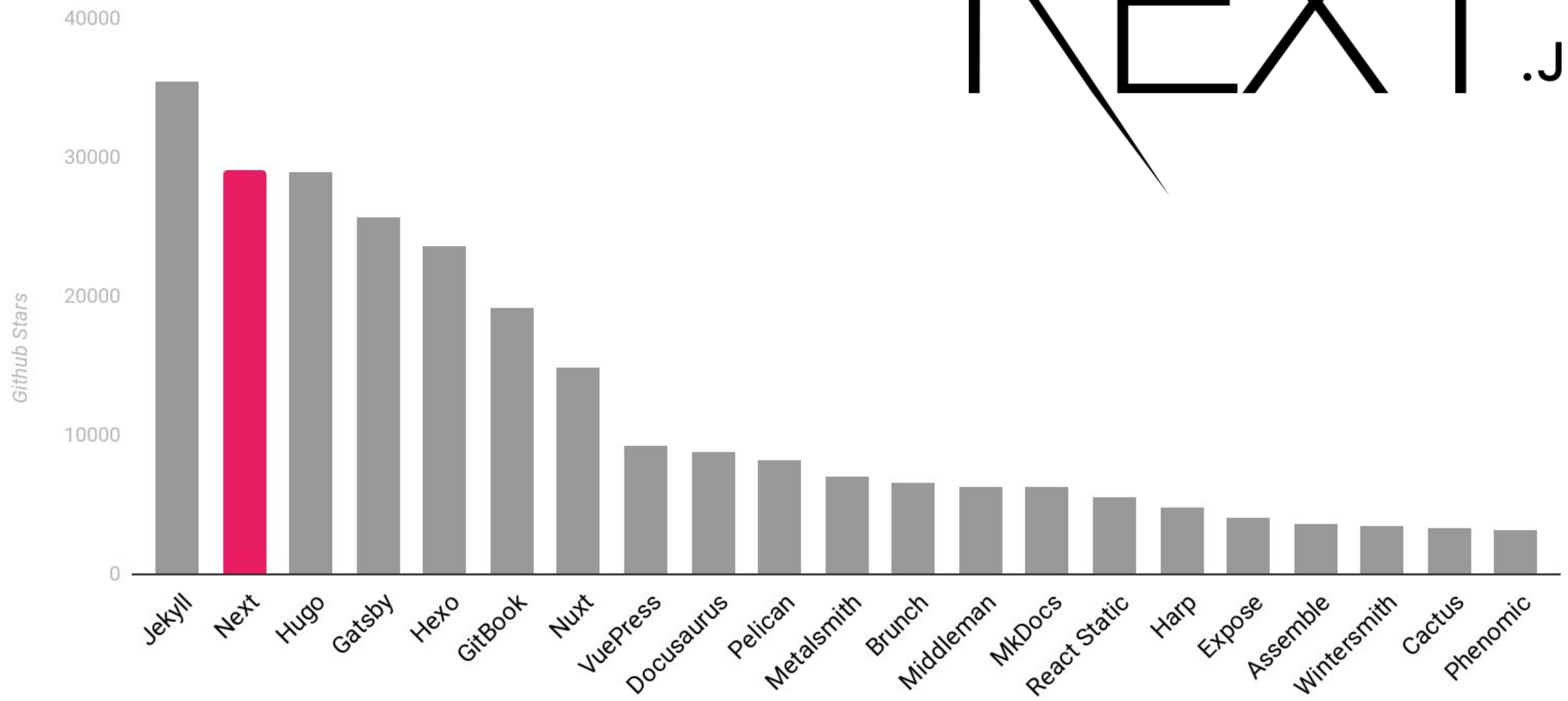
GitHub Pages

Gatsby

Why not use a static site generator?

Static Site Generators



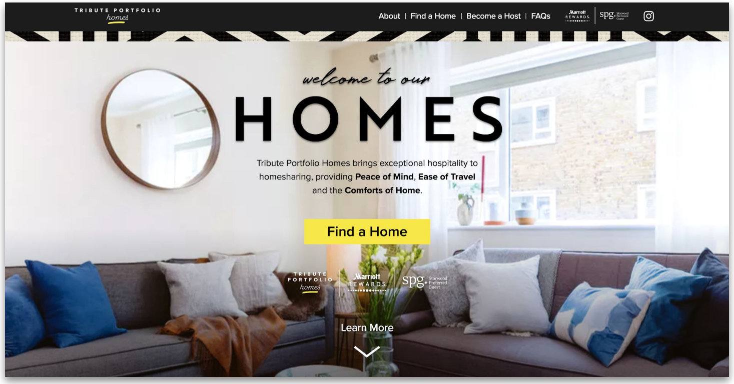

Github Stars
40000
30000
20000
10000
0
Jekyll
Next
Hugo
Gatsby
Hexo
GitBook
Nuxt
VuePress
Docusaurus
Pelican
Metalsmith
Brunch
Middleman
MkDocs
React Static
Harp
Expose
Assemble
Wintersmith
Cactus
Phenomic

NEXT.JS

NEXT.JS

"Next.js is a lightweight framework for static and server-rendered applications."

- Originally for Isomorphic JavaScript Apps
- `next export` - prebuilt apps
- Hey, we're already using this!



Tribute Portfolio Homes brings exceptional hospitality to homesharing, providing **Peace of Mind**, **Ease of Travel** and the **Comforts of Home**.

NEXT.JS

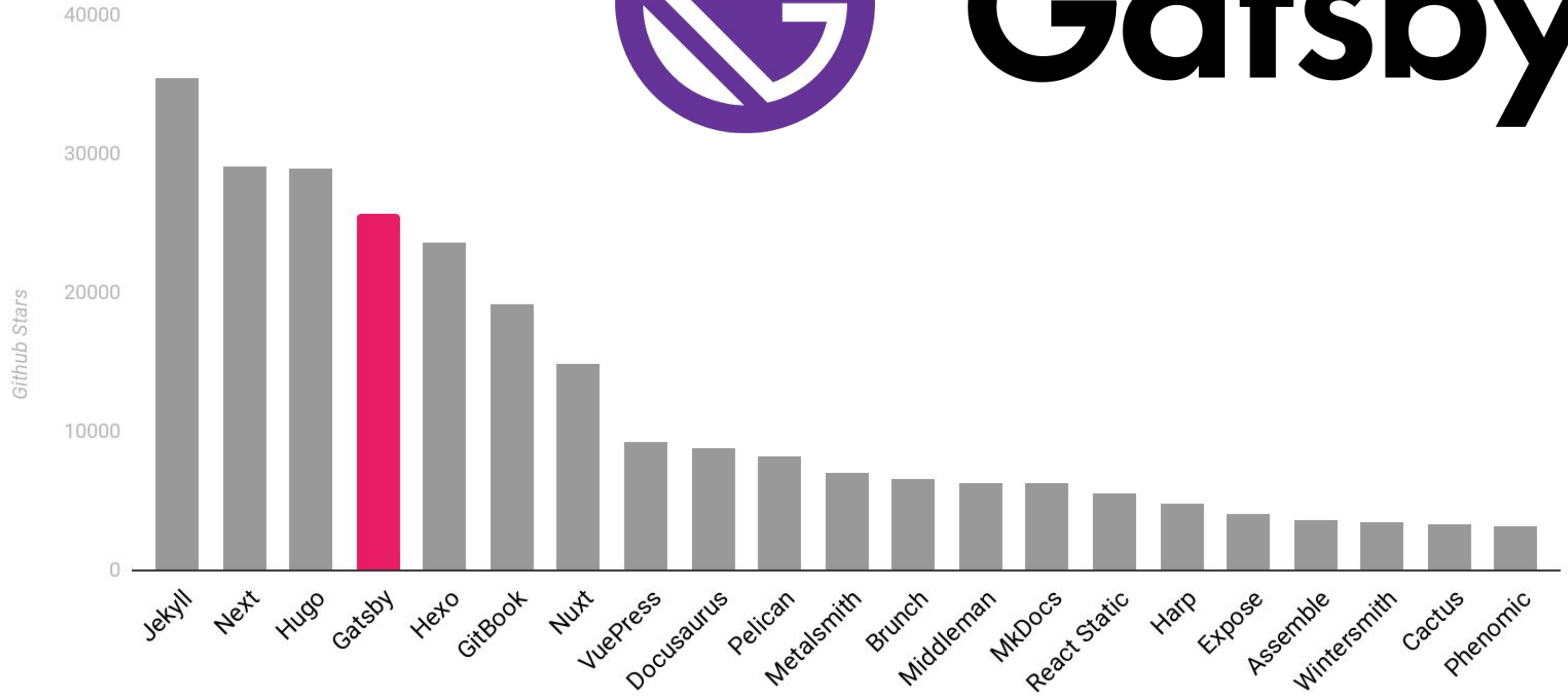
- Great framework
- Well documented
- Large community

But...

- Great **framework** - Next.js more than React
- Predefined structure
- Slow and difficult to optimise

We were looking for something...

- ✓ React based
  - Just like all of our other products
- ⬚ Flexible
  - Had to integrate with our existing codebase
  - Had to support iterative migration
- ⬚ Simple
  - We were finishing our migration to React
  - Did not want new tools (i.e. GraphQL) to be a barrier
- ⬚ Long term
  - Simplicity === we can change fetching strategies, add caching, etc.


40000
30000
20000
10000
0
Github Stars
Jekyll
Next
Hugo
Gatsby
Hexo
GitBook
Nuxt
VuePress
Docusaurus
Pelican
Metalsmith
Brunch
Middleman
MkDocs
React Static
Harp
Expose
Assemble
Wintersmith
Cactus
Phenomic

Gatsby

Gatsby

"Gatsby is a blazing fast modern site generator for React."

- The de facto static site generator for React

Gatsby

- Well documented
- Large community
- Fast & Powerful

But...

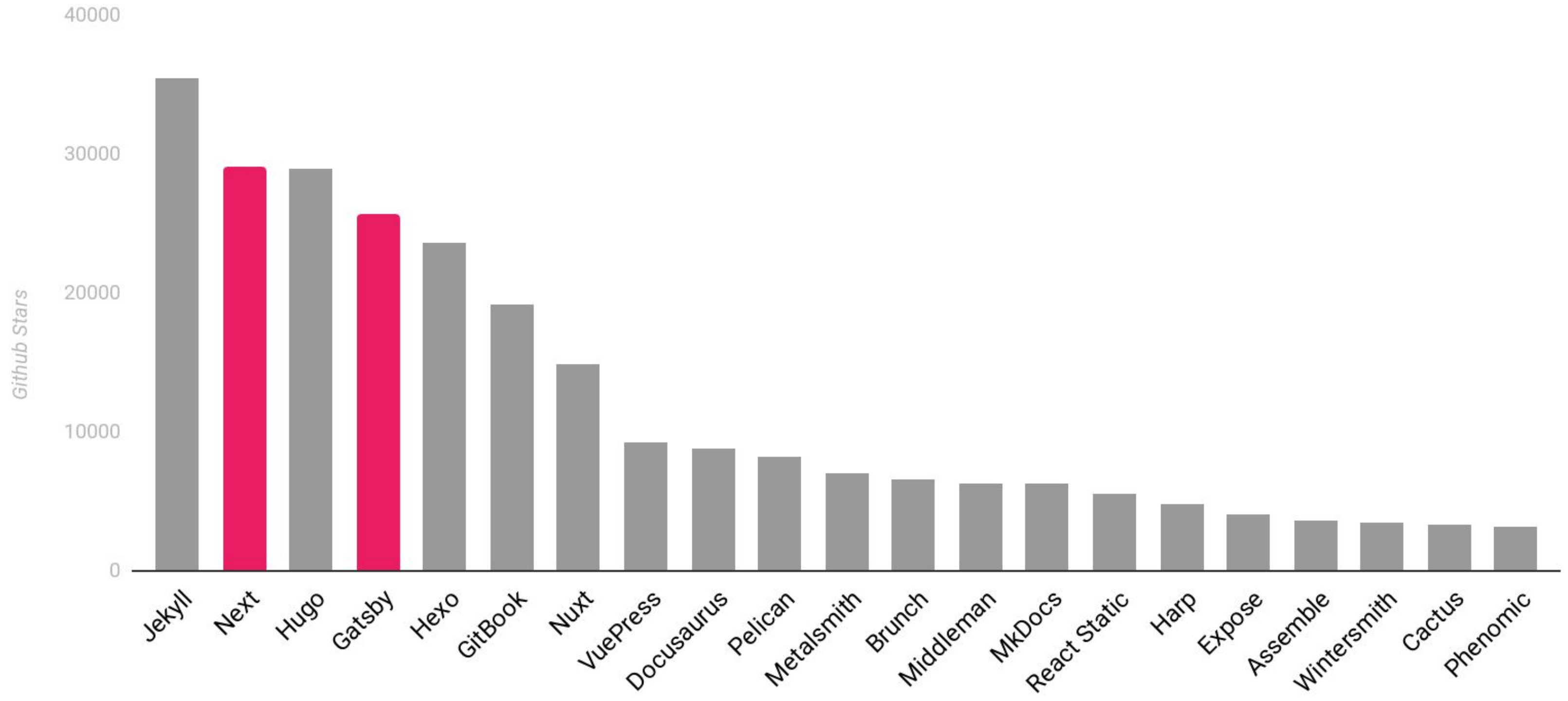
- GraphQL everywhere
- Plugins for everything
- Need to write a plugin to use custom sources

We were looking for something...

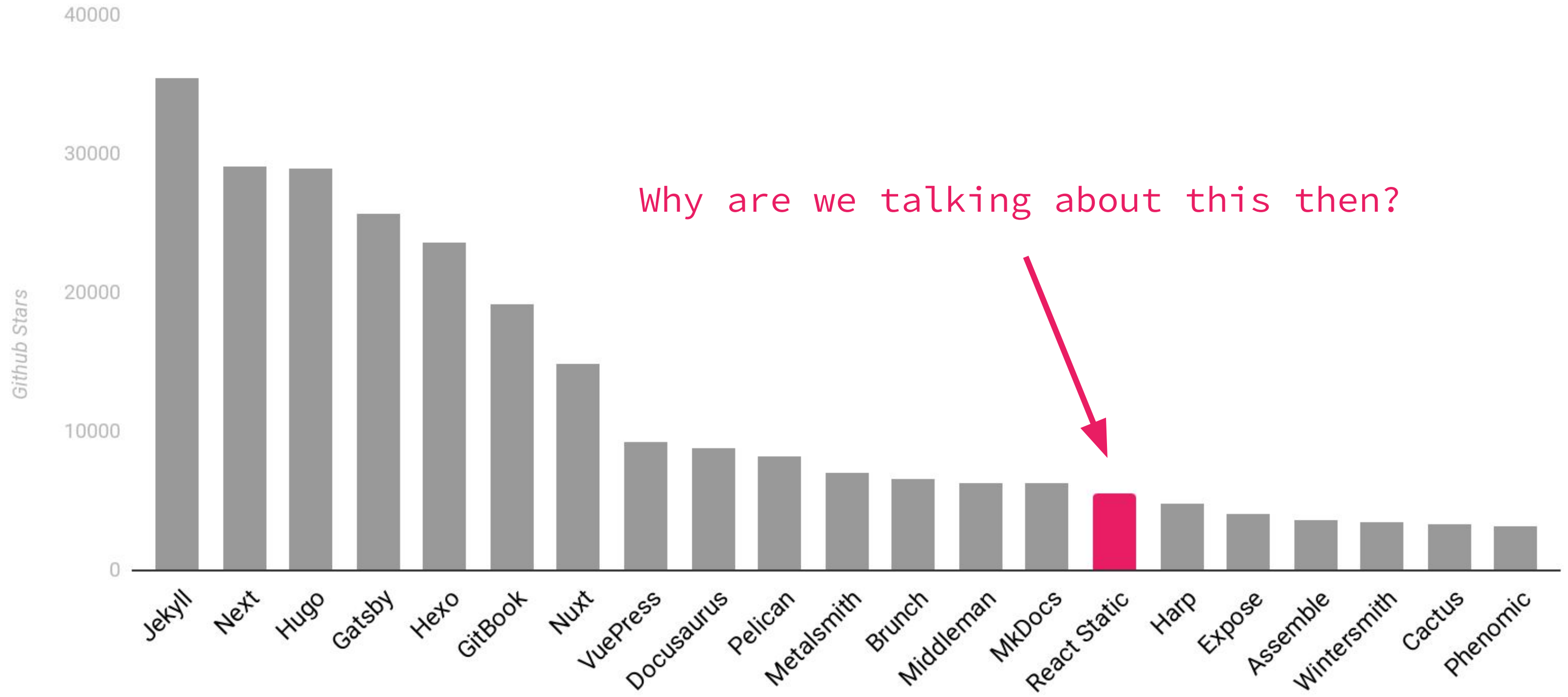
Gatsby

- [✓] React based
  - Just like all of our other products
- [✓] Flexible
  - Had to integrate with our existing codebase
  - Had to support iterative migration
- [ ] Simple
  - We were finishing our migration to React
  - Did not want new tools (i.e. GraphQL) to be a barrier
- [✓] Long term
  - Simplicity === we can change fetching strategies, add caching, etc.

Static Site Generators - two clear options



Static Site Generators - React Static



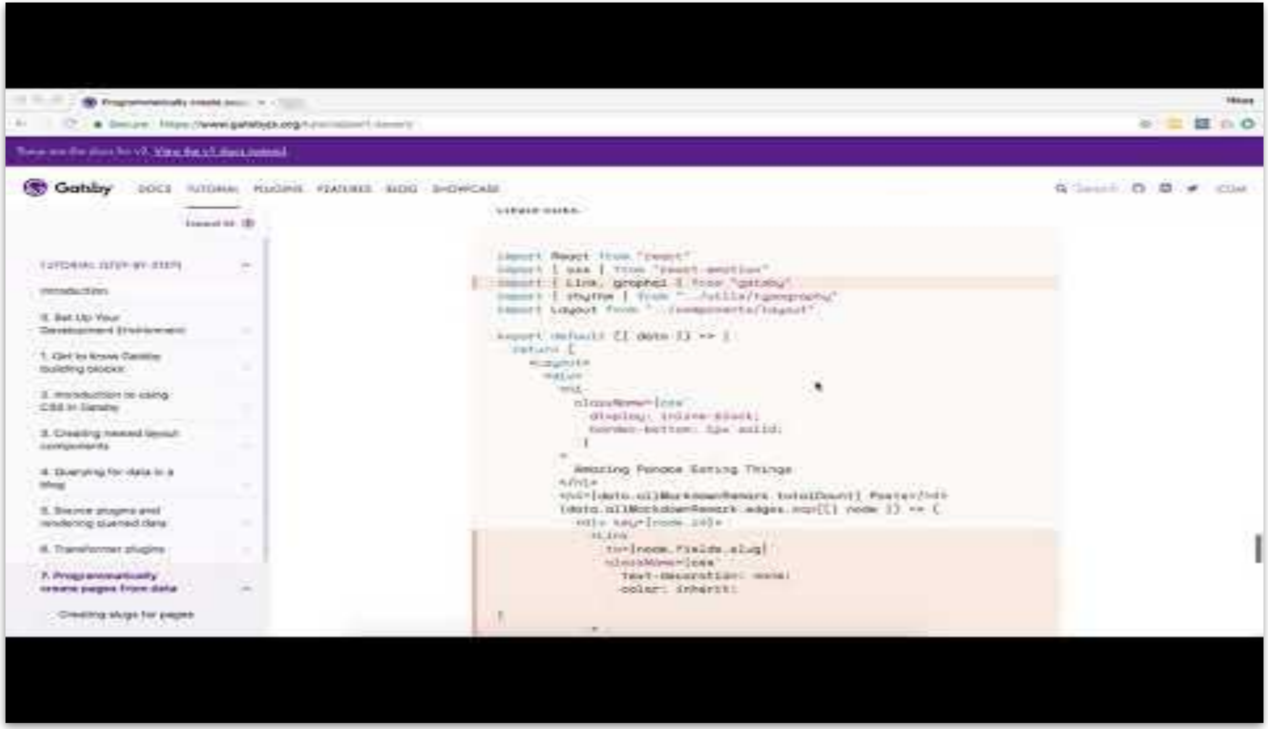
Why are we talking about this then?

React Static

"React-Static is a **fast**, **lightweight**, and **powerful** framework for building static-progressive React applications and websites."

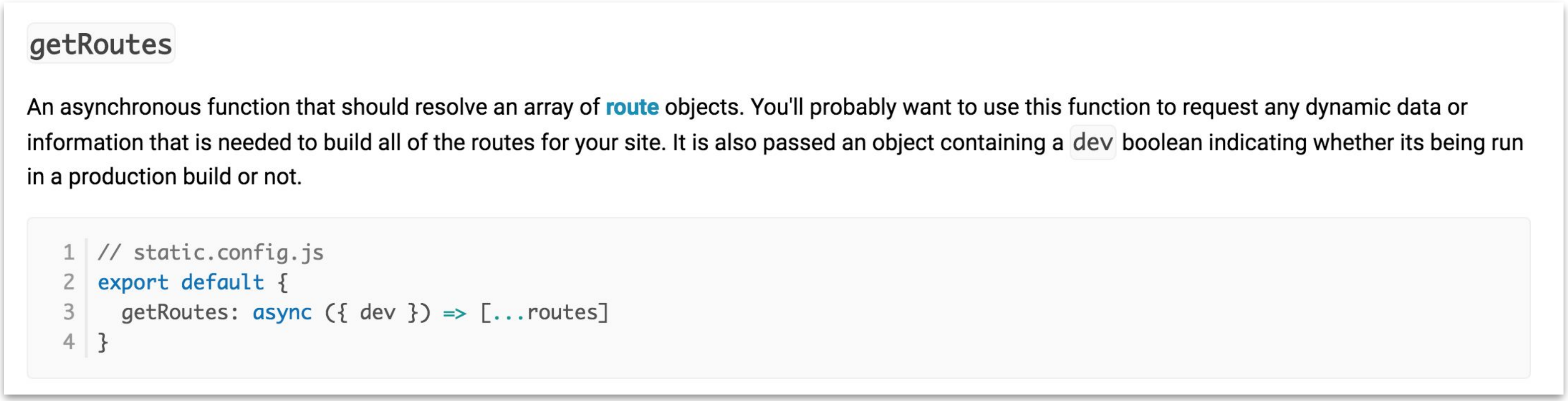
- The promise of the **power of Gatsby without the complexity**
- It's **just React**

Gatsby - "Programatically create pages from data"


Gatsby
DOCS
TUTORIAL
PLUGINS
FEATURES
BLOG
SHOWCASE
TUTORIAL (STEP-BY-STEP)
Introduction
0. Set Up Your Development Environment
1. Get to know Gatsby building blocks
2. Introduction to using CSS in Gatsby
3. Creating nested layout components
4. Querying for data in a blog
5. Source plugins and rendering queried data
6. Transformer plugins
7. Programmatically create pages from data
Creating slugs for pages

```
import React from "react"
import { css } from "react-emotion"
import { Link, graphql } from "gatsby"
import { rhythm } from "../utils/typography"
import Layout from "../components/layout"

export default ({ data }) => {
  return (
    <Layout>
      <div>
        <h1
          className={css`
            display: inline-block;
            border-bottom: 1px solid;
          `}
        >
          Amazing Pandas Eating Things
        </h1>
        <h4>{data.allMarkdownRemark.totalCount} Posts</h4>
        {data.allMarkdownRemark.edges.map(({ node }) => (
          <div key={node.id}>
            <Link
              to={node.fields.slug}
              className={css`
                text-decoration: none;
                color: inherit;
```

React Static - just like, getRoutes(), lol



getRoutes

An asynchronous function that should resolve an array of route objects. You'll probably want to use this function to request any dynamic data or information that is needed to build all of the routes for your site. It is also passed an object containing a dev boolean indicating whether its being run in a production build or not.

```
// static.config.js
export default {
  getRoutes: async ({ dev }) => [...routes]
}
```

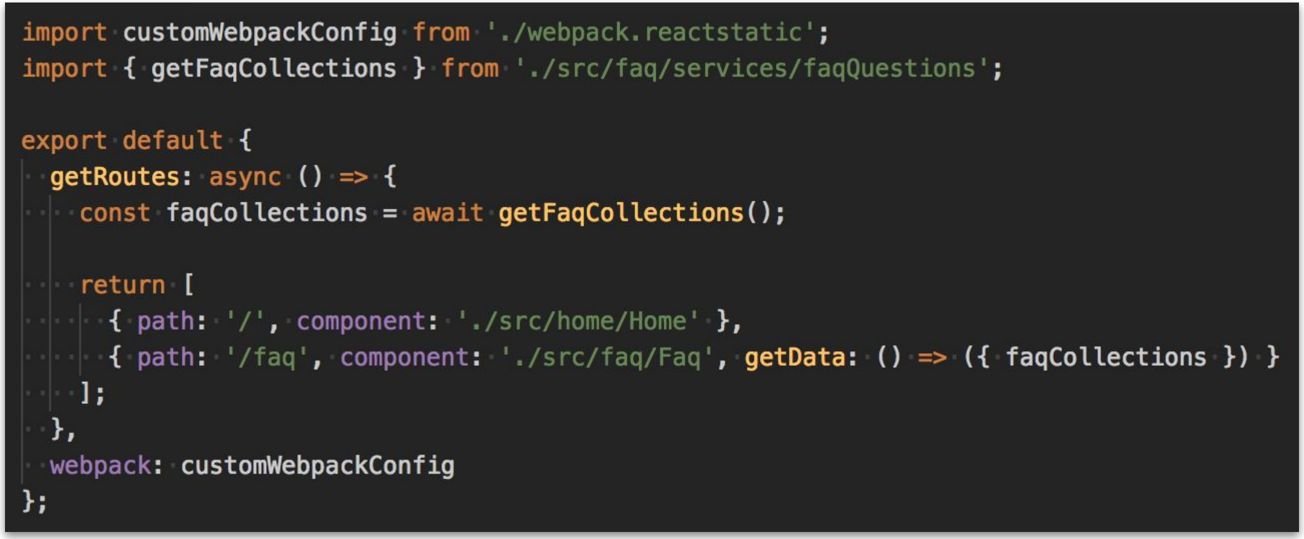
How it works - 2 main concepts

```
/src
    /components
    index.js
    App.js
package.json
static.config.js
```

How it works - static.config.js (1)

- static.config.js
- App.js

Routes

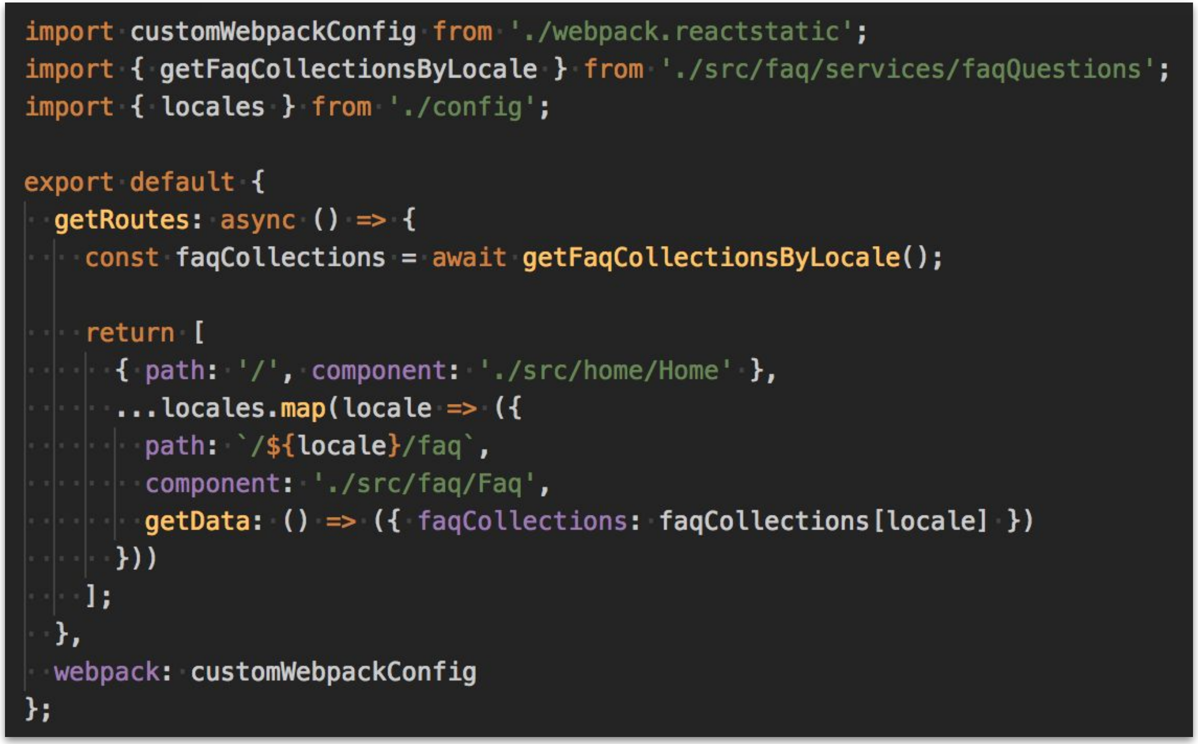


```
import customWebpackConfig from './webpack.reactstatic';
import { getFaqCollections } from './src/faq/services/faqQuestions';

export default {
  getRoutes: async () => {
    const faqCollections = await getFaqCollections();

    return [
      { path: '/', component: './src/home/Home' },
      { path: '/faq', component: './src/faq/Faq', getData: () => ({ faqCollections }) }
    ];
  },
  webpack: customWebpackConfig
};
```

How it works - static.config.js (2)

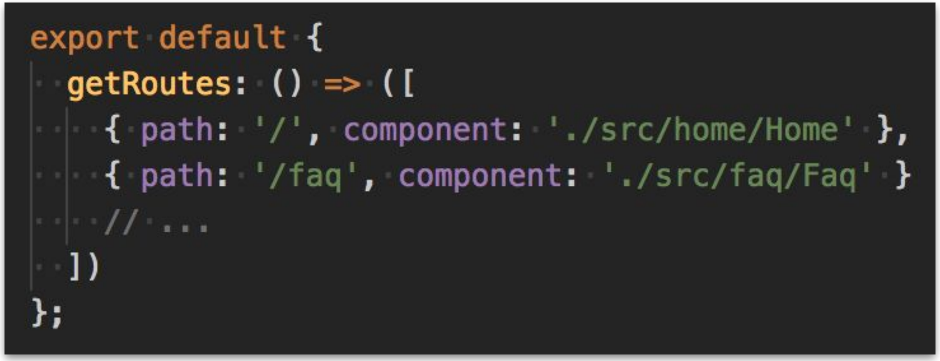
- static.config.js
- App.js

Generate a route for each locale



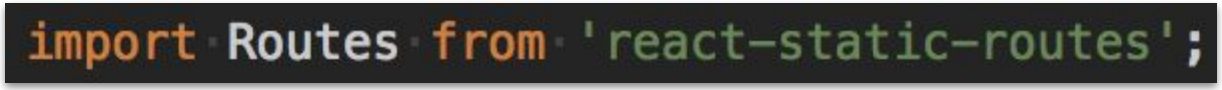
```
import customWebpackConfig from './webpack.reactstatic';
import { getFaqCollectionsByLocale } from './src/faq/services/faqQuestions';
import { locales } from './config';

export default {
  getRoutes: async () => {
    const faqCollections = await getFaqCollectionsByLocale();

    return [
      { path: '/', component: './src/home/Home' },
      ...locales.map(locale => ({
        path: `/${locale}/faq`,
        component: './src/faq/Faq',
        getData: () => ({ faqCollections: faqCollections[locale] })
      }))
    ];
  },
  webpack: customWebpackConfig
};
```

static.config.js



```
export default {
  getRoutes: () => ([
    { path: '/', component: './src/home/Home' },
    { path: '/faq', component: './src/faq/Faq' }
    // ...
  ])
};
```

App.js



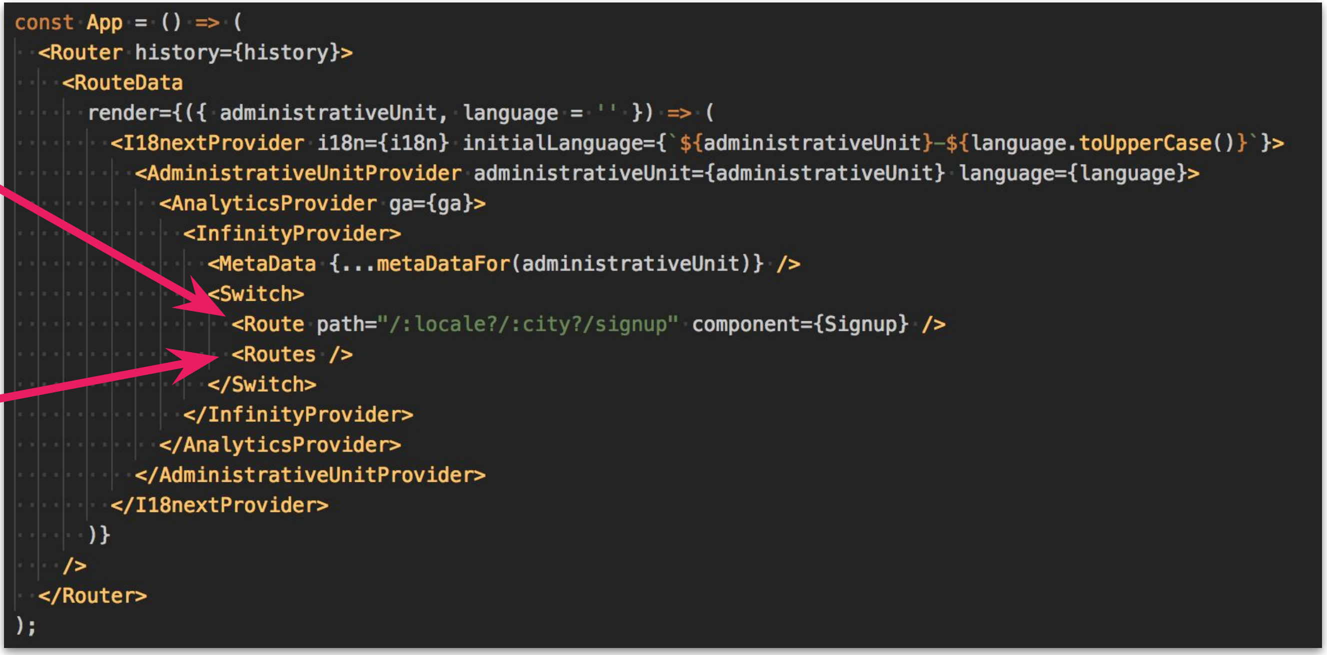
```
import Routes from 'react-static-routes';
```

How it works - App.js

- static.config.js
- App.js

Dynamic routes (Signup)

React-static routes



```
const App = () => (
  <Router history={history}>
    <RouteData
      render={({ administrativeUnit, language = '' }) => (
        <I18nextProvider i18n={i18n} initialLanguage={`${administrativeUnit}-${language.toUpperCase()}`}>
          <AdministrativeUnitProvider administrativeUnit={administrativeUnit} language={language}>
            <AnalyticsProvider ga={ga}>
              <InfinityProvider>
                <MetaData {...metaDataFor(administrativeUnit)} />
                <Switch>
                  <Route path="/:locale?/:city?/signup" component={Signup} />
                  <Routes />
                </Switch>
              </InfinityProvider>
            </AnalyticsProvider>
          </AdministrativeUnitProvider>
        </I18nextProvider>
      )}
    />
  </Router>
);
```

We were looking for something...

React Static

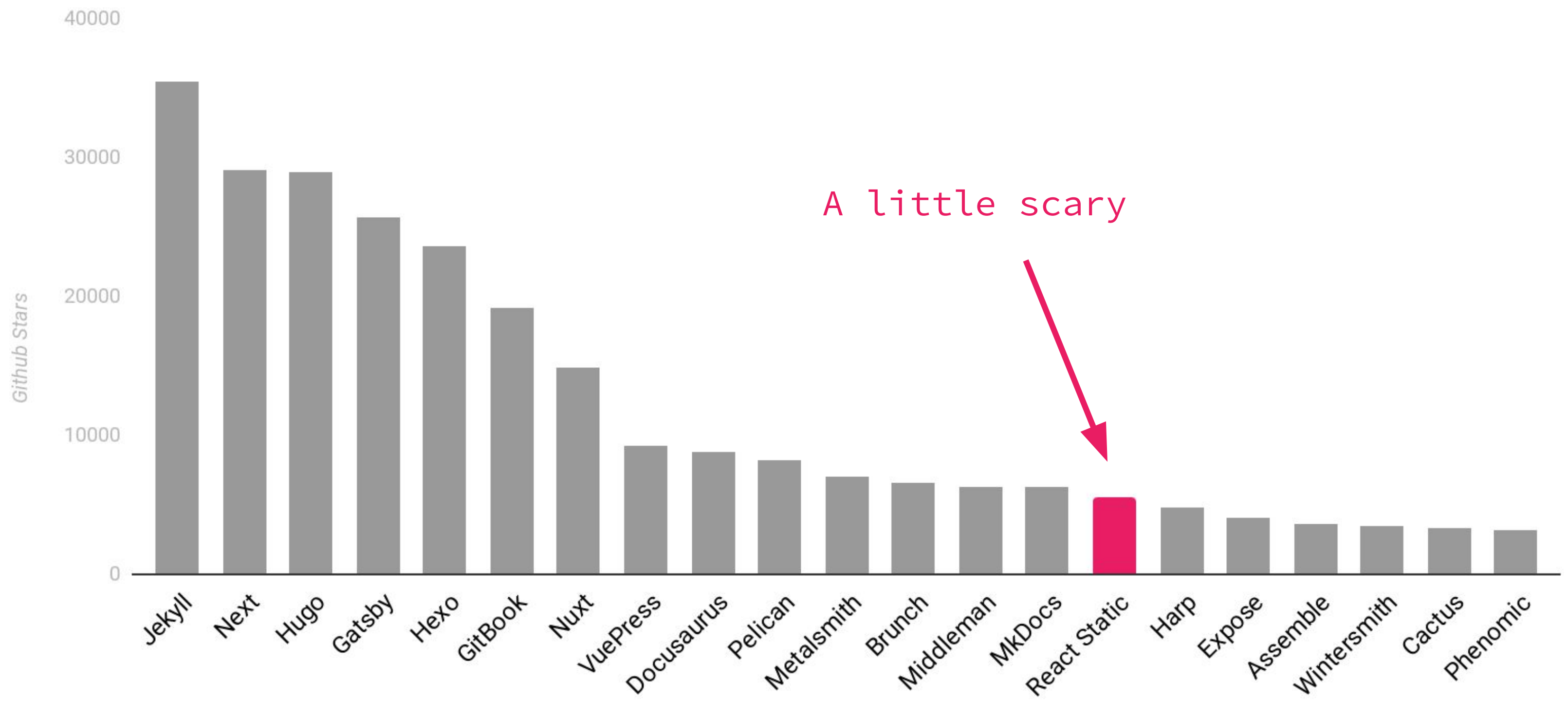
✓ React based
- Just like all of our other products

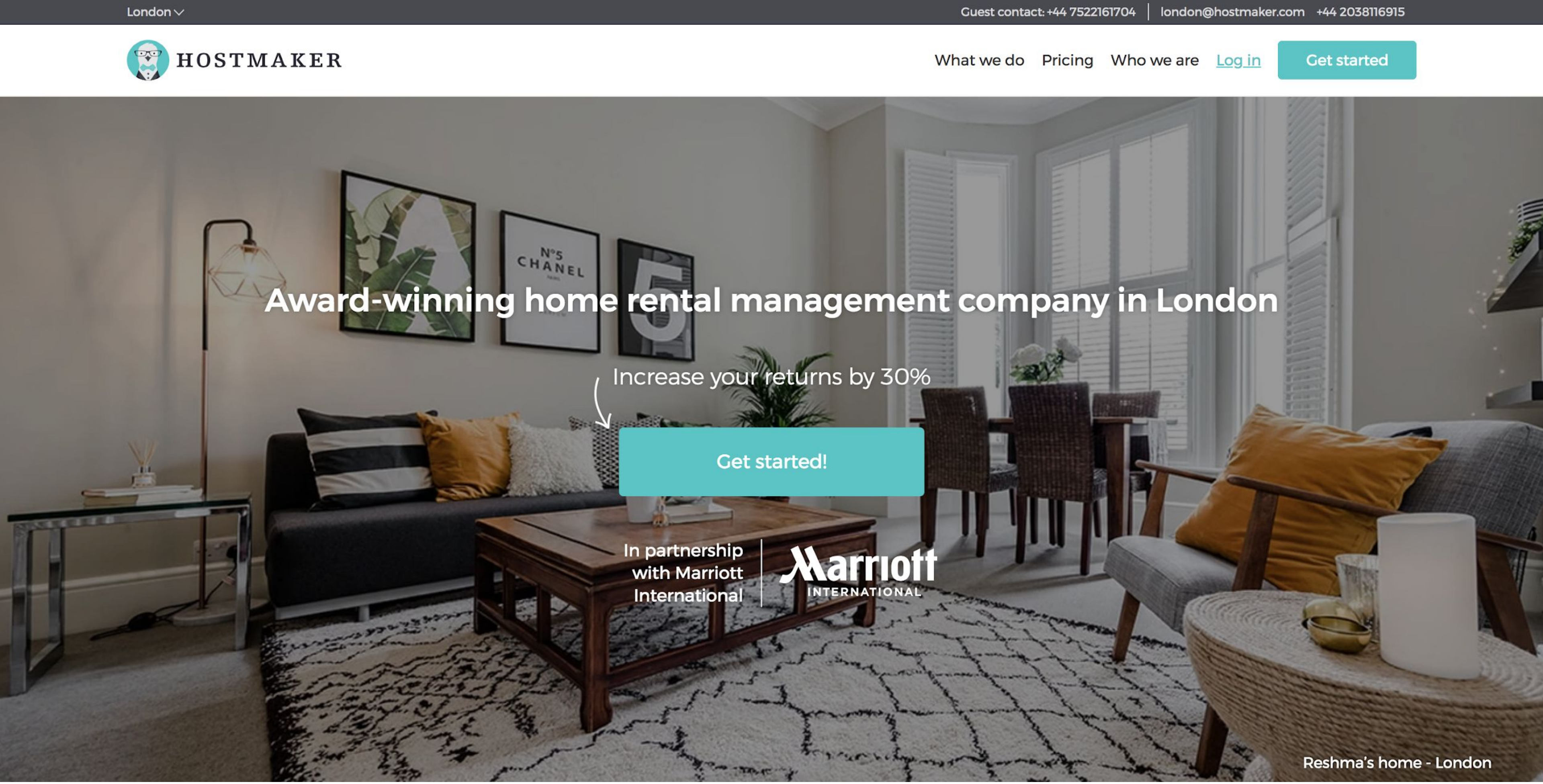
✓ Flexible
- Had to integrate with our existing codebase
- Had to support iterative migration

✓ Simple
- We were finishing our migration to React
- Did not want new tools (i.e. GraphQL) to be a barrier

✓ Long term
- Simplicity === we can change fetching strategies, add caching, etc.

The drawback





London

Guest contact: +44 7522161704 | london@hostmaker.com +44 2038116915

HOSTMAKER

What we do Pricing Who we are Log in Get started

Award-winning home rental management company in London

Increase your returns by 30%

Get started!

In partnership with Marriott International | Marriott INTERNATIONAL

Reshma's home - London

What went well

What went well - Very customisable

- Integrated pretty well with our existing codebase
- Tailored to our needs:
  - CSS modules
  - Imgix
  - Translations and localized content
  - Custom webpack config

What went well - Great development experience

- Hot reloading
- Detailed logging when something goes wrong

What went well - Great performance

- Pretty great out of the box
- Gives us a lot of options long term
  - Code splitting
  - Dynamic loading

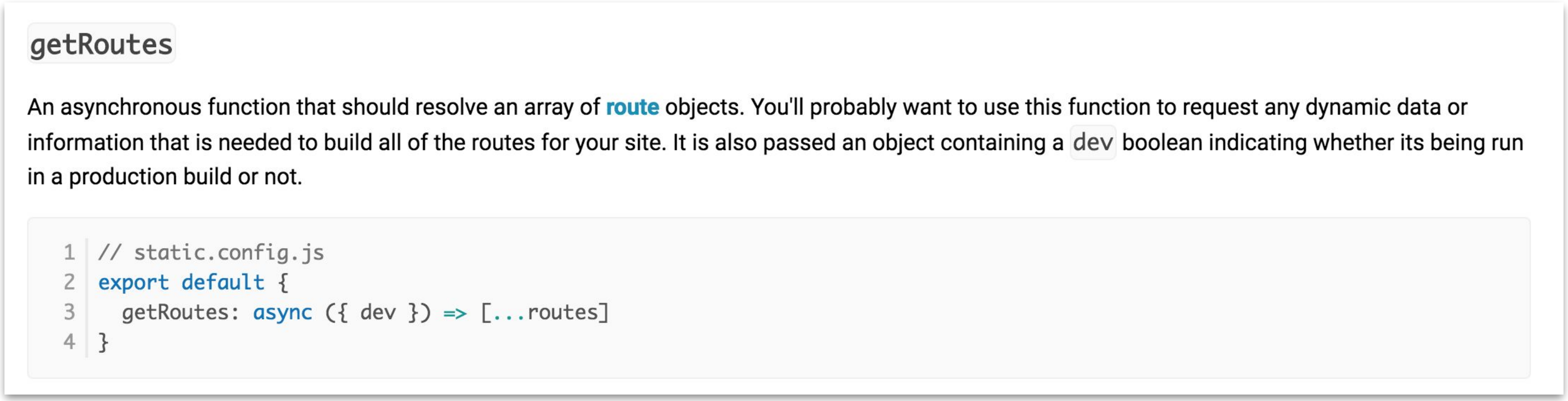
What went well - Did everything it had to

- Started with 500 pre-built pages, now up to 1500+
- Loading data from 12+ endpoints at build time
- Around 3 minute build time
- Can scale long term

What did not go well

What did not go well - Documentation

- Covers everything, but we often had to read the source
- On the bright side, the source is easy to understand



`getRoutes`

An asynchronous function that should resolve an array of route objects. You'll probably want to use this function to request any dynamic data or information that is needed to build all of the routes for your site. It is also passed an object containing a `dev` boolean indicating whether its being run in a production build or not.

```
// static.config.js
export default {
  getRoutes: async ({ dev }) => [...routes]
}
```

What did not go well - Client/Server is still a thing

- Even though there's no "server", there's still a build step run via Node
- Can't always use the obvious solution

Main Takeaway

Stay in React-land

Main Takeaway - getLocale()

- Need to get the current locale (city, language) for rendering data
- Used everywhere
- One of the most important parts of “plumbing”

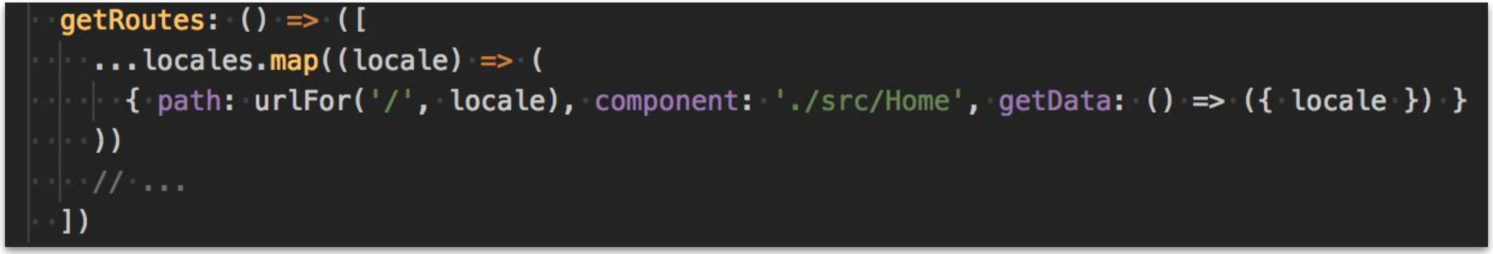
Main Takeaway - standard approach

`localeService.getLocale()`

- Works great in dev and for regular users
- Does **not** work when pre-building

- **The bug is hidden at dev time**
  - All pre-built files use the default fallback
  - Gets the correct value during runtime

Main Takeaway - possible solution?



```
getRoutes: () => ([
    ...locales.map((locale) => (
      { path: urlFor('/', locale), component: './src/Home', getData: () => ({ locale }) }
    ))
    // ...
])
```

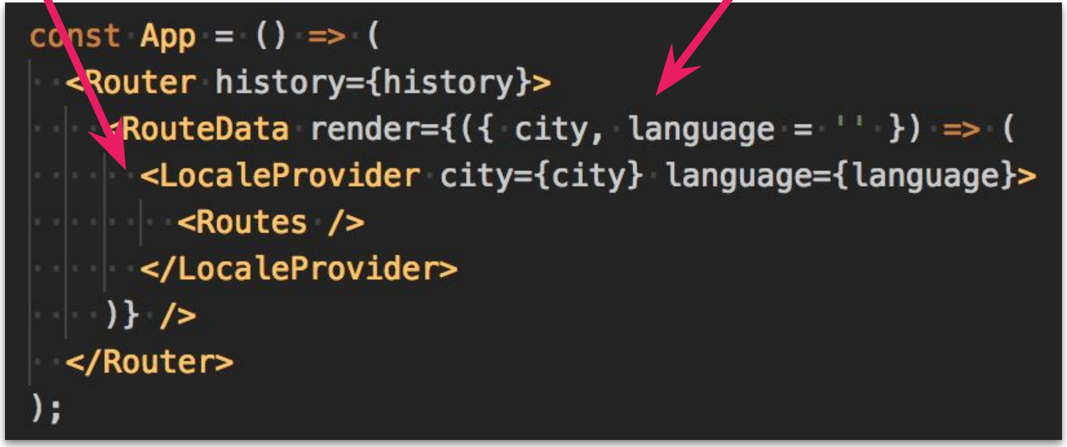
- Need some data globally
- Need to be able to instantiate via props

why not use React Context?

Main Takeaway - the React-land approach (1)

A context provider

From getRoutes()

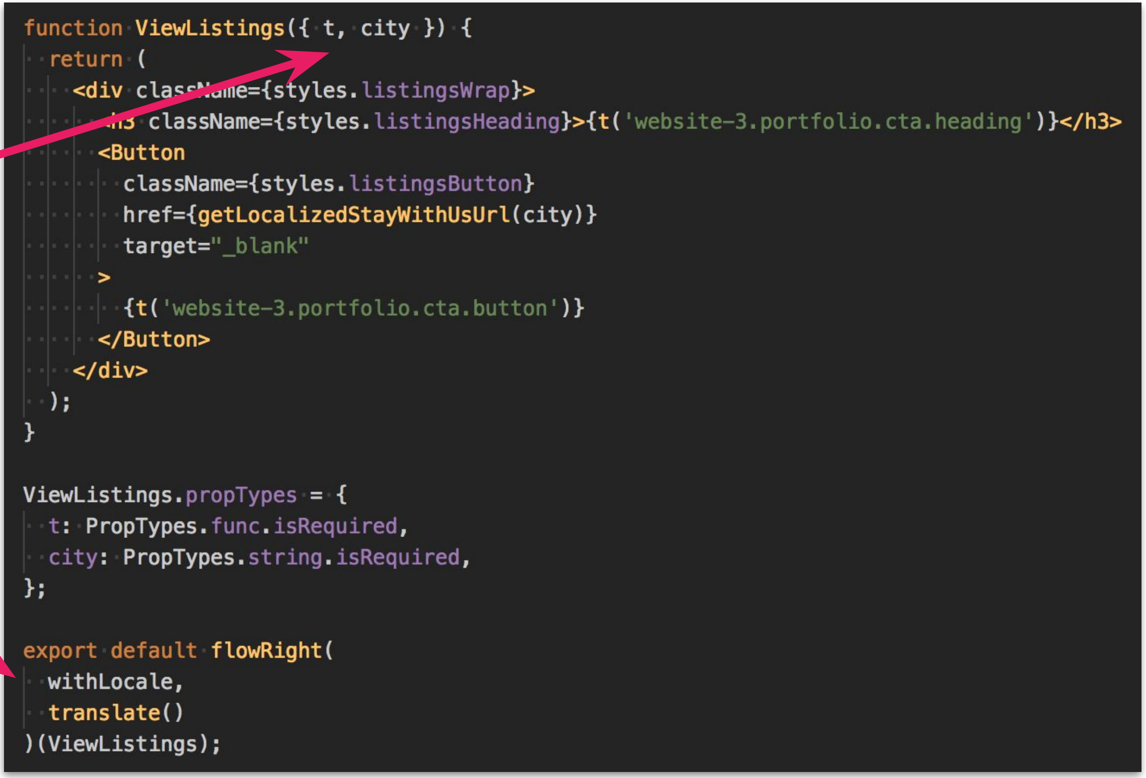


```
const App = () => (
  <Router history={history}>
    <RouteData render={({ city, language = '' }) => (
      <LocaleProvider city={city} language={language}>
        <Routes />
      </LocaleProvider>
    )} />
  </Router>
);
```

Main Takeaway - the React-land approach (2)

Injected via props

HoC wrapper



```
function ViewListings({ t, city }) {
  return (
    <div className={styles.listingsWrap}>
      <h3 className={styles.listingsHeading}>{t('website-3.portfolio.cta.heading')}</h3>
      <Button
        className={styles.listingsButton}
        href={getLocalizedStayWithUsUrl(city)}
        target="_blank"
      >
        {t('website-3.portfolio.cta.button')}
      </Button>
    </div>
  );
}

ViewListings.propTypes = {
  t: PropTypes.func.isRequired,
  city: PropTypes.string.isRequired,
};

export default flowRight(
  withLocale,
  translate()
)(ViewListings);
```

React-land everywhere

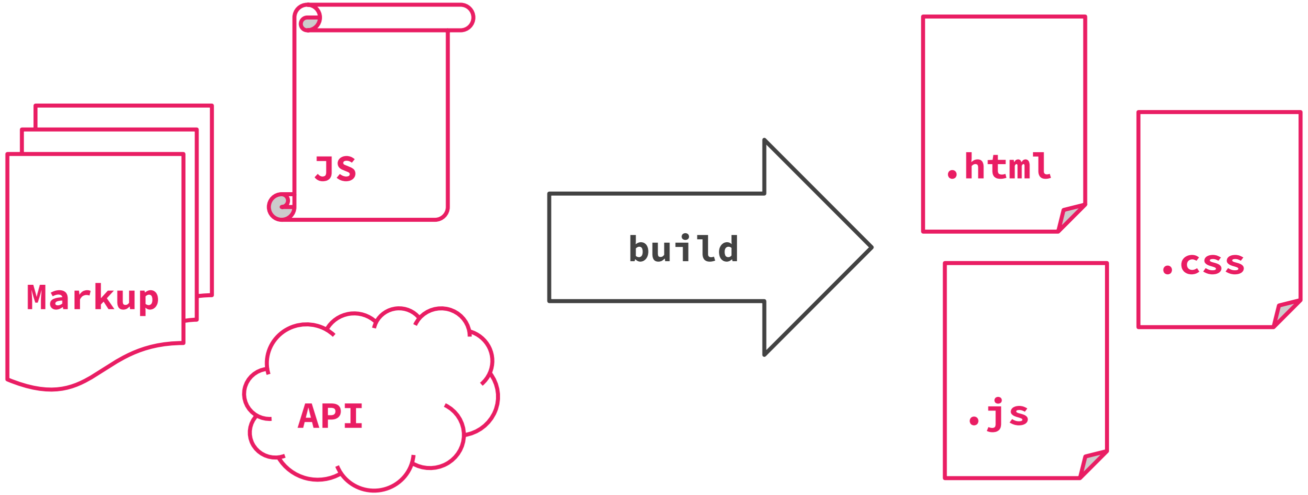
- No more mismatches between build and runtime
- We've since started using providers and context in other projects too

A Different Kind of Context

JAMstack

"Modern web development architecture based on client-side **JavaScript**, reusable **APIs**, and prebuilt **Markup**."

- jamstack.org



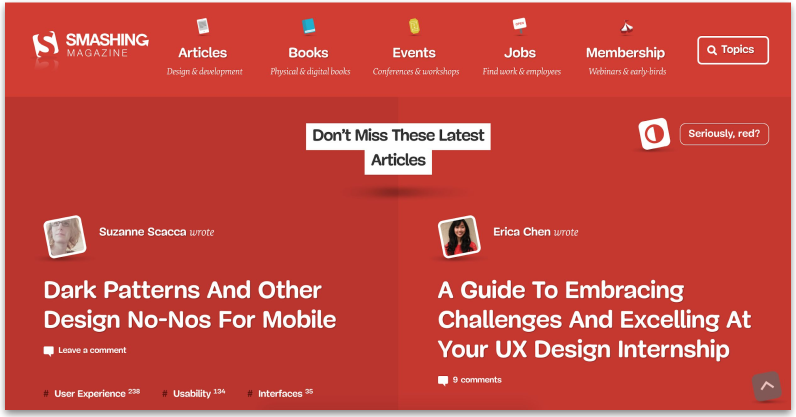
JAMstack Benefits

- Easy deployment
  - Just upload everything to a CDN (i.e. AWS S3)
- Blazing fast
  - Scaling -> just "add more CDN"
  - No waiting for content, no loading screens
- Great dev experience
  - Can't beat free - Github Pages, Netlify etc..
  - Reproducible builds -> easy debugging

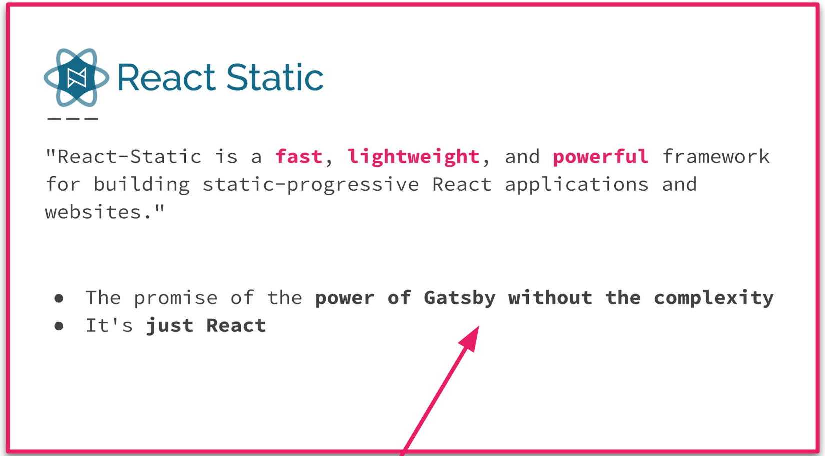
We are not alone

"SmashingMagazine.com is now much faster, they went from **800 ms** time to first load to **80ms**."

- www.netlify.com/case-studies/smashing/



In summary



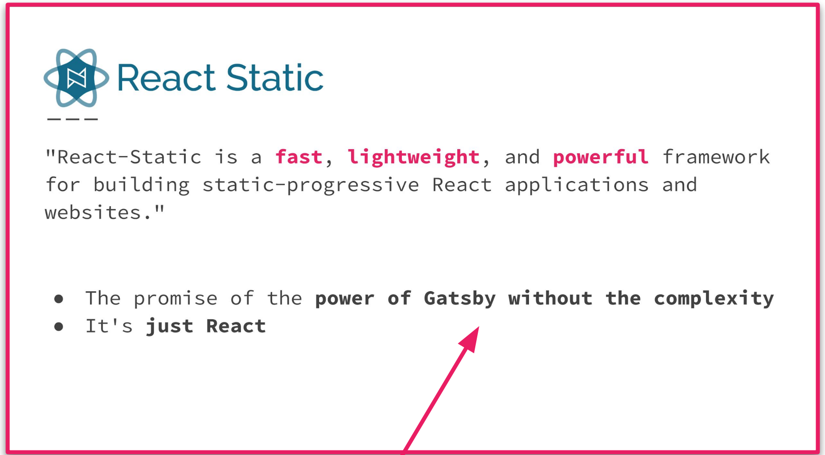
React Static

"React-Static is a **fast**, **lightweight**, and **powerful** framework for building static-progressive React applications and websites."

- The promise of the **power of Gatsby without the complexity**
- It's **just React**

- Mostly
- Would we do it again?

In summary



React Static

"React-Static is a **fast**, **lightweight**, and **powerful** framework for building static-progressive React applications and websites."

- The promise of the **power of Gatsby without the complexity**
- It's **just React**

- Mostly
- Would we do it again?

Yes!

Thank you!

Live Demo @
hostmaker.com/careers

Let's keep in touch:

nikas.praninskas.com

github.com/nikaspran

@nikaspran