

# getting started with Nuxt and static sites



Debbie O'Brien



Frontend Tech Lead  
@Patterson Agency  
Mentor @OpenClassrooms  
Slack Moderator @Treehouse  
Contributor @webpack



debs\_obrien

# Why Nuxt?

Nuxt is a wrapper built on top of webpack and Vue

# Why Nuxt?

Why do I need a framework on top of a framework?

# Why Nuxt?

For SSR you can just use `vue-server-renderer`

# Why Nuxt?

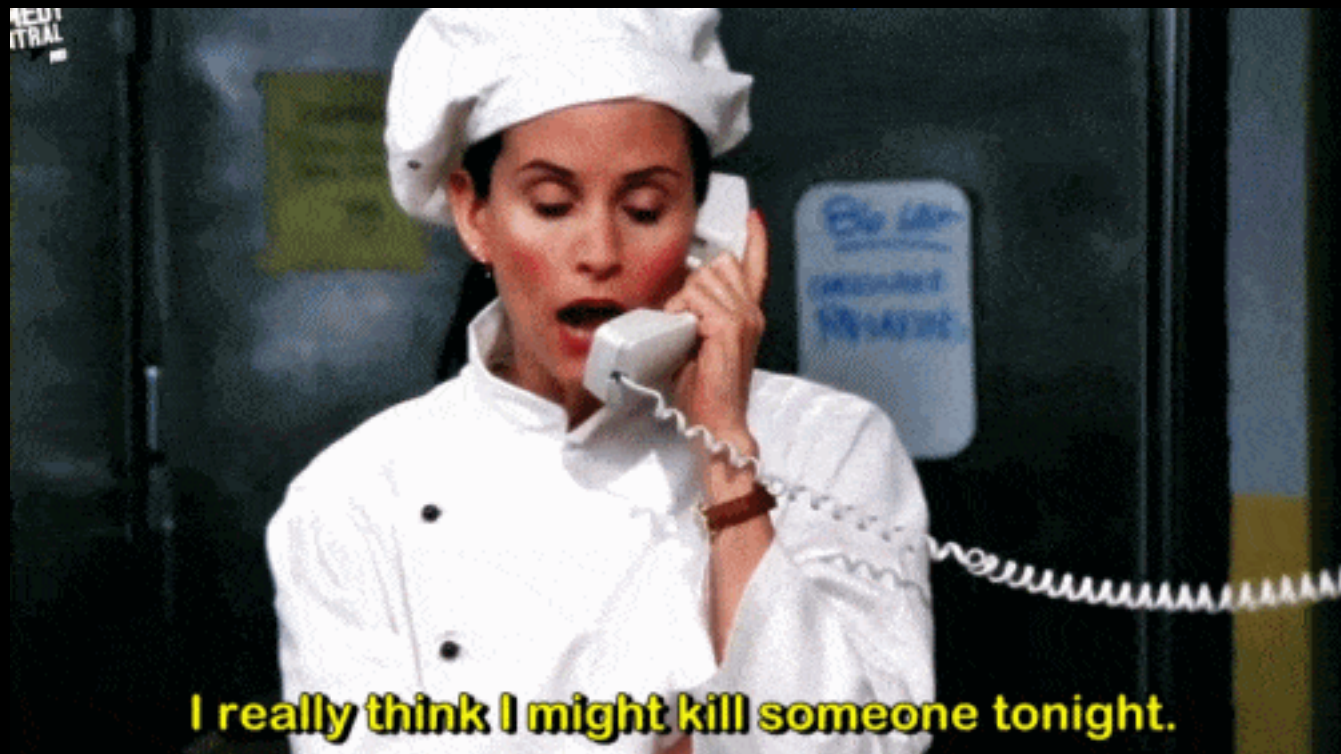
Implementing it was hard

# Why Nuxt?

I needed dynamic pages

# Why Nuxt?

What do you do when you need help?



**I really think I might kill someone tonight.**



You call a Vue Core Team  
Member

Eduardo San Martín





# Why Nuxt?

Why do I need a framework on top of a framework?

# Why Nuxt?

Because Eduardo said so

# Why Nuxt?

Because the Core team members of Vue are recommending Nuxt

Why should You use  
Nuxt?

# Why Nuxt?

Nuxt helps you write universal apps more simply

# Why Nuxt?

Code splitting of routes out of the box

Dynamic pages and routes

Generate static sites

SEO

...

# Why Nuxt?

Nuxt makes it easy to create Universal Apps

# Universal App

# Universal App

A universal app is code that can execute on both  
client and server

SPA

# SPA

No page refreshes but.....

Many crawlers find it difficult to parse and understand your website especially if getting your data from an api

# Universal App

# Universal App

Preloads your application on a server

sends rendered HTML to the browser for every route

Improves SEO

Makes loading happen faster

# Universal App

You get:

`<title> <meta> <head> <h1> <body>` etc

before any javascript is loaded

This helps crawlers determine what's on the page

# Static Sites

# Static Sites

It will generate HTML for every one of your routes

Routes are pre-rendered

# Static Sites

Page loads faster

Helps search engines and social media crawlers

No server needed

Everything gets generated during the build phase

# Static Sites

Aren't we going back in time? Back to the 90's!

Why?

# Static Sites

The 90's was a cool era

Especially for web development



[Privacy Policy](#) [Cookie Settings](#) [Terms](#) [Accessibility](#) [AdChoices](#)

SPACE JAM, characters, names, and all related  
indicia are trademarks of Warner Bros. © 1996



You've made it: Jam Central Station, the central depository for all things Space Jam. From the best seats in the house, you can peruse the production notes, find out about the filmmakers, check out the theatrical trailer, and look at a bunch of photos from the film.

SPACE JAM, characters, names, and all related indicia are trademarks of Warner Bros. © 1996



# Static Sites

It's not a 90's static site

# Static Sites

When a user visits your static website, the **hydration process kicks** in and your application becomes a **single page application**.

# Static Sites

What's Hydration?

nick@nite



# Static Sites

Hydration is:



# Static Sites

It makes a static site act like a SPA

All the Spa benefits with all the Static site benefits

# Static Sites

It's a match made in heaven

# Static Sites

Universal rendering without the need of a server

You can host on  
github, netlify, Amazon S3 etc

# Static Sites v SSR Site

It's faster => pre-generated static assets

It's cheaper => no server needed

It's Safer => Less prone to attacks



# Static Sites

Static generated / pre rendered means **pre render** the application **once during build** and **serve it as a SPA**



Nuxt.js

@nuxt\_js

Following



For all of you who struggle with the difference between Server Side Rendering and Static Site Generation:

Static Site Generation === Server Side Rendering *\*only\** at build time



12:07 AM - 21 Feb 2019

# Patterson Travel Static Site

# Patterson.travel

A live example of a Nuxt static site

Hola, somos patterson.travel, el copiloto de los viajes de tu marca. La agencia que diseña, coordina y gestiona tus viajes y eventos de empresa. Para el día a día o para ocasiones especiales.

La agencia que diseña, coordina y gestiona tus viajes y eventos de empresa. Para el día a día o para ocasiones especiales.

for your business

Servicios

[Ir a Servicios >](#)



Elements Console Sources Network Performance Memory Audits >>

11:09:29 AM - patterson.trave

Performance 100 Accessibility 88 Best Practices 93 SEO 100

Score scale: 90-100 50-89 0-49

There were issues affecting this run of Lighthouse:

- Chrome extensions negatively affected this page's load performance. Try auditing the page in incognito mode or from a Chrome profile without extensions.

Performance 100

Metrics

First Contentful Paint	0.3 s	First Meaningful Paint	0.3 s
Speed Index	0.4 s	First CPU Idle	0.8 s
Time to Interactive	0.8 s	Estimated Input Latency	20 ms

View Trace

Values are estimated and may vary.

Diagnostics

More information about the performance of your application.

Console What's New Search Coverage Rendering

top Filter Default levels

\*\* dataslayer: injecting Launch monitors \*\* VM468:2

WebExtension::executeScript: content script injected VM421:1

# Patterson.travel

The frontend team on using a Nuxt static site:



# Patterson.travel

The marketing team on using a nuxt static site:



# Best practices: Nuxt and SEO

# Best practices: Nuxt and SEO

Nuxt uses vue-meta to handle the `<head>` element in each of your pages.

There are 3 ways to set up the head element:

# Best practices: Nuxt and SEO

- 1) Default meta tags for all your pages. You can set this in the nuxt config
- 2) Setting up meta tags for your pages individually by defining a head element
- 3) Setting up meta tags for your dynamic pages

# Dynamic Routes

# Dynamic Routes

Dynamic routes are defined by prefixing your .vue page in the pages folder with an underscore.

`pages/users/_username.vue`

`pages/blog/_id.vue`

`pages/blog/_slug.vue`

# Dynamic Routes

To generate dynamic routes you need to add a generate option to your nuxt.config.

By default Nuxt won't generate static files for any dynamic routes as it doesn't know what the id/slug will be.

# nuxt.config.js

```
generate: {  
  routes: [  
    '/users/1',  
    '/users/2',  
    '/users/3'  
  ]  
}
```

# Generates an index per route

```
[nuxt] Generating...
  nuxt:render Rendering url / +154ms
  nuxt:render Rendering url /users/1 +12ms
  nuxt:render Rendering url /users/2 +33ms
  nuxt:render Rendering url /users/3 +7ms
  nuxt:generate Generate file: /index.html +21ms
  nuxt:generate Generate file: /users/1/index.html +31ms
  nuxt:generate Generate file: /users/2/index.html +15ms
  nuxt:generate Generate file: /users/3/index.html +23ms
  nuxt:generate HTML Files generated in 7.6s +6ms
[nuxt] Generate done
```

# Dynamic Routes

Wait a Second

What if I have 100 users or blog posts or I don't know what the id or name will be. This isn't very dynamic.



# function that returns a promise

```
generate: {  
  routes: function () {  
    return axios.get('https://my-api/users')  
      .then((res) => {  
        return res.data.map((user) => {  
          return '/users/' + user.id  
        })  
      })  
  }  
},
```

# speed it up with payload

```
generate: {  
  routes: function() {  
    return axios.get('https://my-api/users').then((res) => {  
      return res.data.map((user) => {  
        return {  
          route: '/users/' + user.id,  
          payload: user,  
        }  
      })  
    })  
  },  
},
```

# access the payload from /users/\_id.vue

```
asyncData ({ params, error, payload }) {  
  if (payload) return { user: payload }  
  else return {user: await backend.fetchUser(params.id)}  
}
```

# Nuxt Boilerplate

# Nuxt Boilerplate

Before you start to build:

- Add your linters
- Set up prettier
- Set up your pre commit hooks
- Set up your meta for SEO
- Set up your testing

# prettierrc.js

```
module.exports = {  
  ...  
  printWidth: 80,  
  tabWidth: 2,  
  useTabs: false,  
  semi: false,  
  singleQuote: true,  
  trailingComma: 'es5',  
  bracketSpacing: true,  
  arrowParens: 'always',  
  endOfLine: 'lf'  
}
```

# eslintrc.js

```
extends: [  
  '@nuxtjs',  
  'plugin:nuxt/recommended',  
  'plugin:prettier/recommended',  
  'prettier',  
  'prettier/vue'  
],  
plugins: ['prettier'],  
// add your custom rules here  
rules: {}
```

# eslinttrc.js

```
rules: {
```

```
  // Only allow debugger in development
```

```
  'no-debugger': process.env.PRE_COMMIT ? 'error' : 'off',
```

```
  'no-console': process.env.PRE_COMMIT
```

```
    ? ['error', { allow: ['warn', 'error'] }]
```

```
    : 'off'
```

```
}
```

# stylelint.config.js

```
extends: [  
  'stylelint-config-standard',  
  'stylelint-config-prettier'  
],  
plugins: [  
  // Bring in some extra rules for SCSS  
  'stylelint-scss',  
],
```

# stylelint.config.js

```
rules: {  
  'color-named': 'never',  
  'color-no-hex': true,  
  'no-empty-source': null,  
  'string-no-newline': null  
}
```

# Add Global Sass Variables

Add @nuxtjs/style-resources

Share variables, mixins, functions across all style files (no @import needed)

Support for SASS, LESS and Stylus

# nuxt.config.js

```
modules: [  
  'nuxt-purgecss',  
  '@nuxtjs/style-resources'  
],  
// globally load all our sass variables  
styleResources: {  
  scss: ['@assets/scss/variables.scss'],  
},
```

# PurgeCSS

## nuxt.config.js

```
purgeCSS: {  
  mode: 'postcss',  
},
```

# Pre Commit Hooks: Husky

```
"scripts": {  
  "dev": "nuxt",  
  "build": "nuxt build",  
  "start": "nuxt start",  
  "generate": "nuxt generate",  
  "stylelint": "stylelint \"./**/*.vue,scss\"",  
  "lint": "eslint --ext .js,.vue --ignore-path .gitignore ."  
},  
"husky": {  
  "hooks": {  
    "pre-commit": "CROSS-ENV PRE_COMMIT=1 npm run lint && npm run stylelint"  
  }  
},
```

# Deploying to github

# nuxt.config.js

```
const routerBase =  
  process.env.DEPLOY_ENV === 'GH_PAGES'  
    ? {  
      router: {  
        base: '/get-started-with-nuxt/'  
      }  
    }  
    : {}  
  
module.exports = Object.assign(routerBase, {  
  mode: 'universal',  
})
```

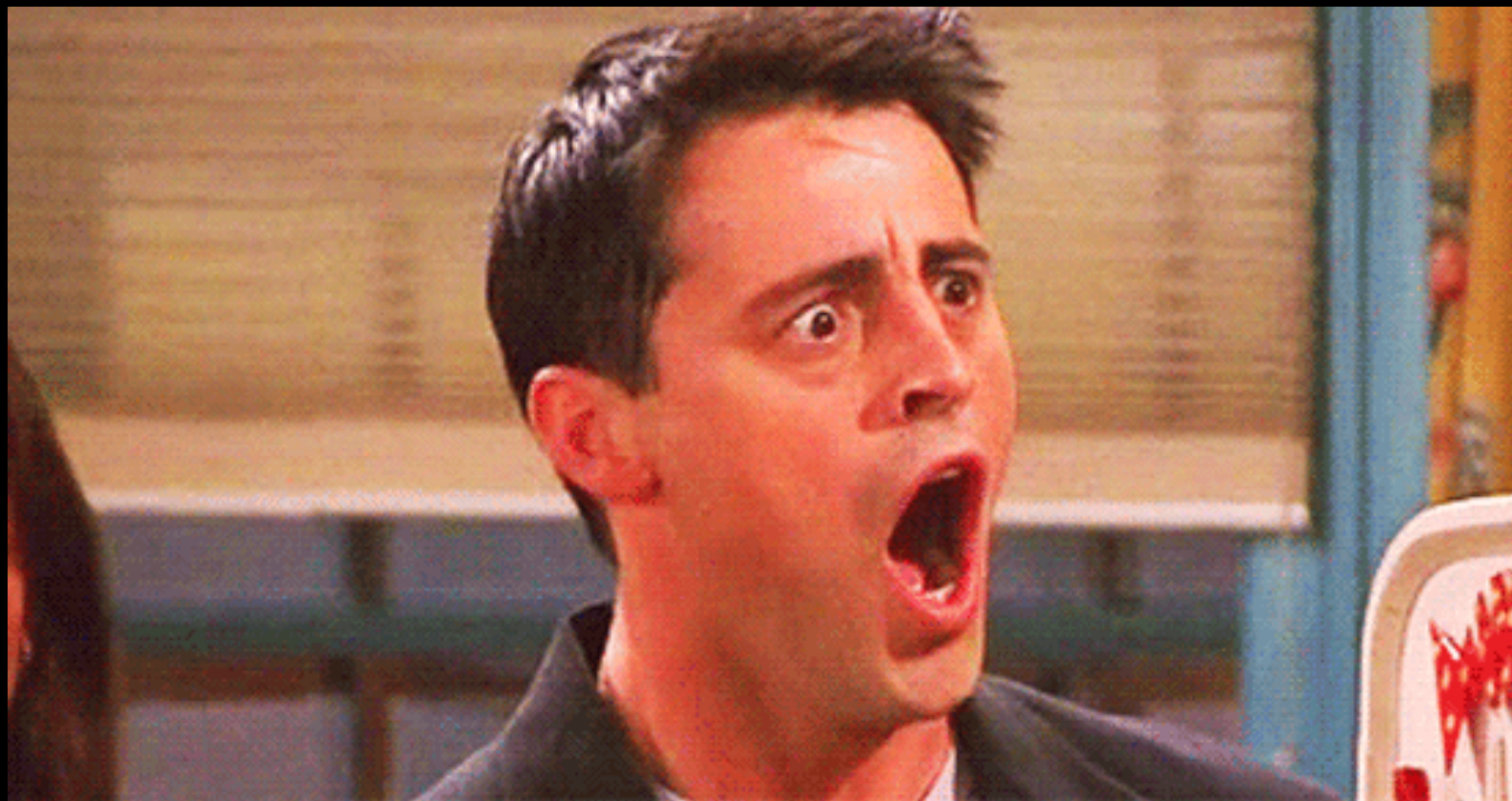
# package.json

```
"generate:gh-pages": "DEPLOY_ENV=GH_PAGES nuxt generate",  
"deploy": "push-dir --dir=dist --branch=gh-pages --cleanup",
```

# terminal

```
npm run generate:gh-pages
```

```
npm run deploy
```



# Nuxt Benefits

# AsyncData

Fetches data before the page is sent to the client

you can “sync” your data or define data that you want rendered on the server side

This is then merged with the component data

# AsyncData

At the moment:  
asyncData can only be applied to components in the  
pages directory.

# Code Splitting

Every route gets its own JavaScript file with just the code that is needed to run on that route.

# Page Transitions

Each page gets wrapped in a `<transition>` element

Easily create transitions between pages

# css/scss file

```
.page-enter-active,  
.page-leave-active {  
| transition: opacity 0.4s;  
}
```

```
.page-enter,  
.page-leave-active {  
| opacity: 0;  
}
```

# Layout Transitions

```
.layout-enter-active,  
.layout-leave-active {  
|   transition: opacity 0.4s;  
| }  
}
```

```
.layout-enter,  
.layout-leave-active {  
|   opacity: 0;  
| }  
}
```

# Page Specific Transitions

```
.about-enter-active,  
.about-leave-active {  
| transition: opacity 0.4s;  
}
```

```
.about-enter,  
.about-leave-active {  
| opacity: 0;  
}
```

# about.vue

```
export default {  
  transitions: 'about'  
}
```

# Why Nuxt

# Why Nuxt?

Nuxt is easy to get started  
great community  
great documentation  
fantastic chat -  
ask questions to core team members

# Why Nuxt?

Follow Nuxt on twitter for the latest up to date news including what companies have started to use Nuxt as their framework of choice



Sébastien Chopin

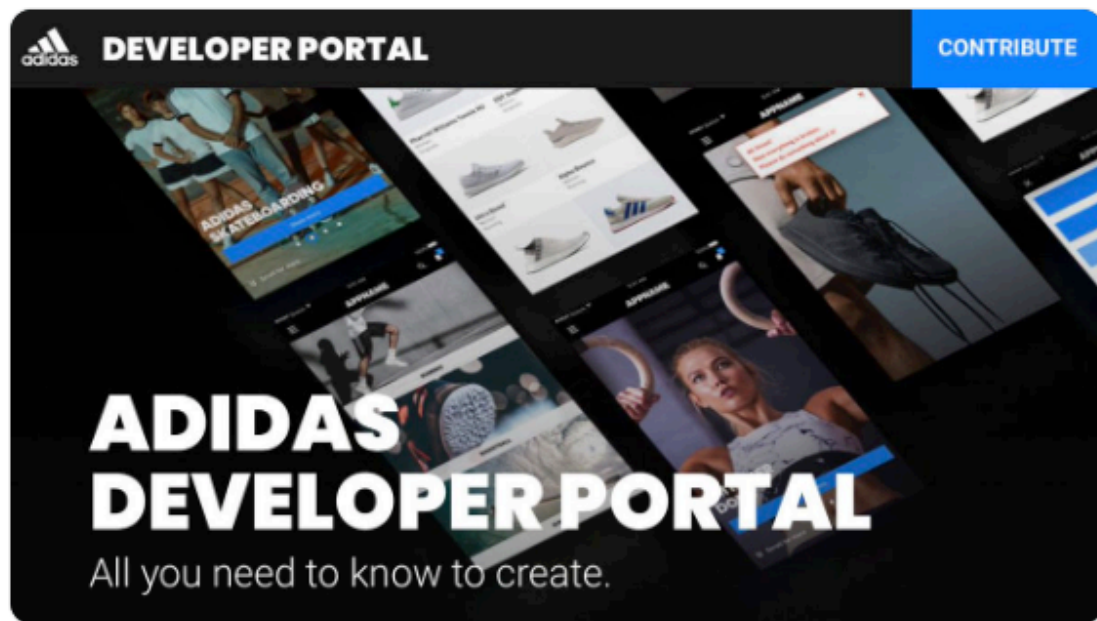
@Atinux

Following



Adidas have chosen [@nuxt\\_js](#) for their  
developer portal 🥰

[adidas.github.io](https://adidas.github.io)



3:39 PM - 17 Mar 2019

# Learn Nuxt



*Are you*  
NUXT?



KTMDAKEWOOOOO.TUMBLR

# Thank you



debs\_obrien