



Nuxt - Introduction



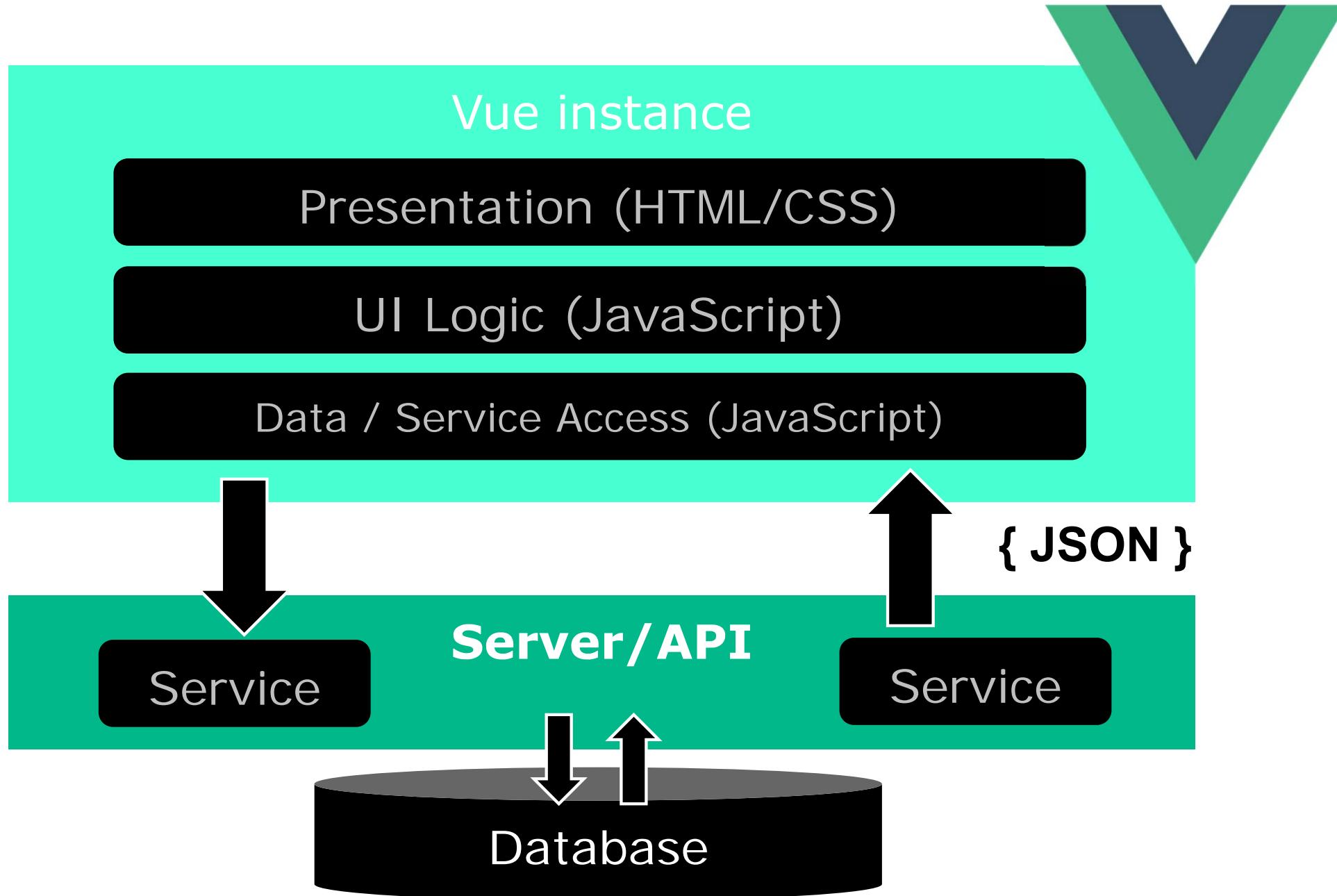
Peter Kassenaar –
info@kassenaar.com



+



Nuxt applications



Using Nuxt

“Nuxt.js presets all the configuration needed to make your development of a Vue.js application enjoyable.”

Why Nuxt?



Nuxt solves problems for you

1. Building from scratch is time consuming and difficult
 - Set up vue-router, vuex, vue-meta, default values...
2. Convention over configuration – default folder structure
 - /assets, /pages, /layouts, etc.
3. Routing configuration can get lengthy/time consuming
 - Vue generates routes from the /pages folder
4. Vue has no standard way for configuration – Nuxt does
 - Use the nuxt.config.js file as a central entry point



5. Vue is not SEO friendly by default

- Nuxt can generate static, indexable files with the Universal configuration option

6. Default Vue apps can potentially load slowly

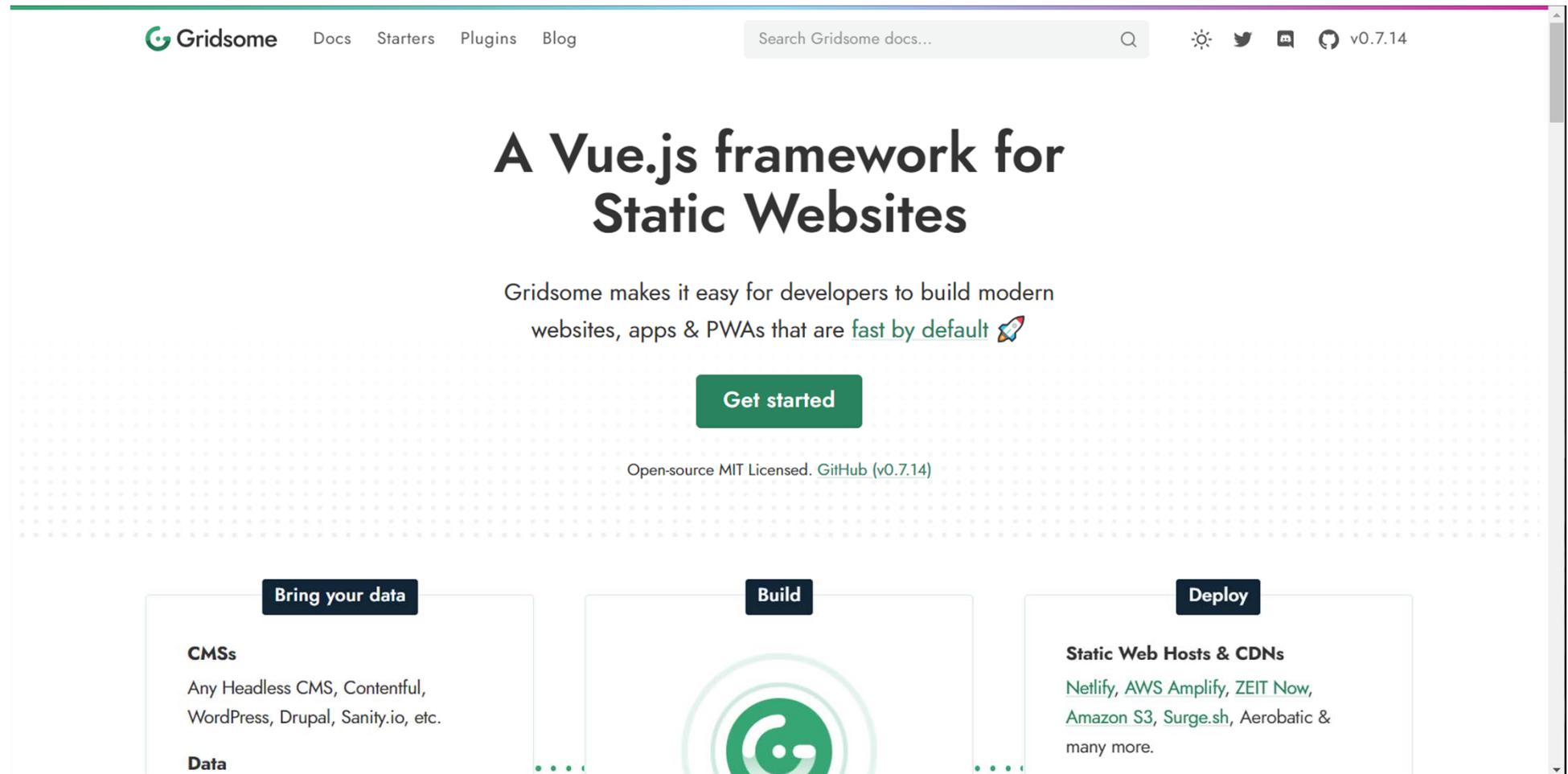
- Nuxt: code splitting, pre-render pages, speed up delivery

7. Difficult to change behavior of standard Vue-applications

- Nuxt: module system that you can apply

Alternatives

There are more static site generators available. Best known: Gridsome



The screenshot shows the Gridsome homepage. At the top, there is a navigation bar with the Gridsome logo, links to Docs, Starters, Plugins, and Blog, a search bar containing "Search Gridsome docs...", and icons for GitHub, Twitter, LinkedIn, and a rocket ship. The main title "A Vue.js framework for Static Websites" is prominently displayed in large, bold, dark font. Below the title, a subtitle reads "Gridsome makes it easy for developers to build modern websites, apps & PWAs that are fast by default" followed by a rocket ship emoji. A green "Get started" button is centered below the subtitle. Further down, it says "Open-source MIT Licensed. GitHub (v0.7.14)". At the bottom, there is a diagram illustrating the Gridsome workflow: "Bring your data" (CMSs like Headless CMS, Contentful, WordPress, Drupal, Sanity.io), "Build" (represented by a circular icon with a stylized 'G'), and "Deploy" (Static Web Hosts & CDNs like Netlify, AWS Amplify, ZEIT Now, Amazon S3, Surge.sh, Aerobatic). The "Data" section also lists "Any Headless CMS, Contentful, WordPress, Drupal, Sanity.io, etc."

<https://gridsome.org/>

If you want to check it out:

The screenshot shows a web browser window displaying an article from VueSchool. The header includes the VueSchool logo, navigation links for 'Tutorials', 'Video Courses', a search icon, and a settings gear icon. The main title of the article is '9 reasons to use Gridsome for your next Vue application', written by Jose Garrido. Below the title is a section titled 'What is Gridsome?'. A paragraph explains that Gridsome is an open source VueJS framework for building static sites, emphasizing its speed and performance features like code splitting and asset optimization. At the bottom of the article page, there are social sharing buttons for Twitter and Facebook, and a link to the next article about Vue 3 Beta.

VueSchool

Tutorials Video Courses Q

9 reasons to use Gridsome for your next Vue application

Written by Jose Garrido

What is Gridsome?

Gridsome is an open source VueJS framework used for building static generated sites. The main focus of Gridsome is that the final website will always be fast, **really fast**. This is achieved by adding performant features such as code splitting, asset optimization, progressive images, and link prefetching by default. With a Gridsome website you'll get almost perfect page speed

[Tweet](#) [Facebook](#)

Up Next:
[Vue 3 Beta is released - Learn the latest news](#)

<https://vueschool.io/articles/vuejs-tutorials/9-reasons-to-use-gridsome-for-your-next-vue-application/>



Creating a Nuxt project

Using the command line tool

Creating a Nuxt project



- Using the scaffolding tool
 - `npx create-nuxt-app` OR
 - `yarn create nuxt-app`
- Answer simple questions from the install prompt

```
npm
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. Alle rechten voorbehouden.

C:\Users\Gebruiker>cd Desktop
C:\Users\Gebruiker\Desktop>npx create-nuxt-app
[##.....] \ fetchMetadata: sill resolveWithNewModule is-object@1.0.1 checking installable status
```

Creating a Nuxt app



You can pick the defaults from the prompt (or pick your own)

```
PS C:\npm
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. Alle rechten voorbehouden.

C:\Users\Gebruiker>cd Desktop

C:\Users\Gebruiker\Desktop>npx create-nuxt-app
npx: installed 379 in 14.247s
> Generating Nuxt.js project in C:\Users\Gebruiker\Desktop
? Project name my-nuxt-app
? Project description My first Nuxt-app
? Use a custom server framework (Use arrow keys)
> none
  express
  koa
  adonis
  hapi
  feathers
  micro
(Move up and down to reveal more choices)
```

Creating a Nuxt App



Pick Universal if you want to SEO-optimize your app!

Server Side Rendering (SSR)

Client Side navigation

```
cmd: npm
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. Alle rechten voorbehouden.

C:\Users\Gebruiker>cd Desktop

C:\Users\Gebruiker\Desktop>npx create-nuxt-app
npx: installed 379 in 14.247s
> Generating Nuxt.js project in C:\Users\Gebruiker\Desktop
? Project name my-nuxt-app
? Project description My first Nuxt-app
? Use a custom server framework none
? Choose features to install Axios
? Use a custom UI framework bootstrap
? Use a custom test framework none
? Choose rendering mode (Use arrow keys)
> Universal ←
    Single Page App
```



Running the sample app

cd into the folder if necessary (like cd my-nuxt-app)

Run a development version:

```
npm run dev
```

Run a build/production version:

```
npm run build
```

```
npm start
```

```
> my-nuxt-app@1.0.0 dev C:\Users\Gebruiker\Desktop\my-nuxt-app
> nuxt

          Nuxt.js v2.6.3
          Running in development mode (universal)

          Listening on: http://localhost:3000/

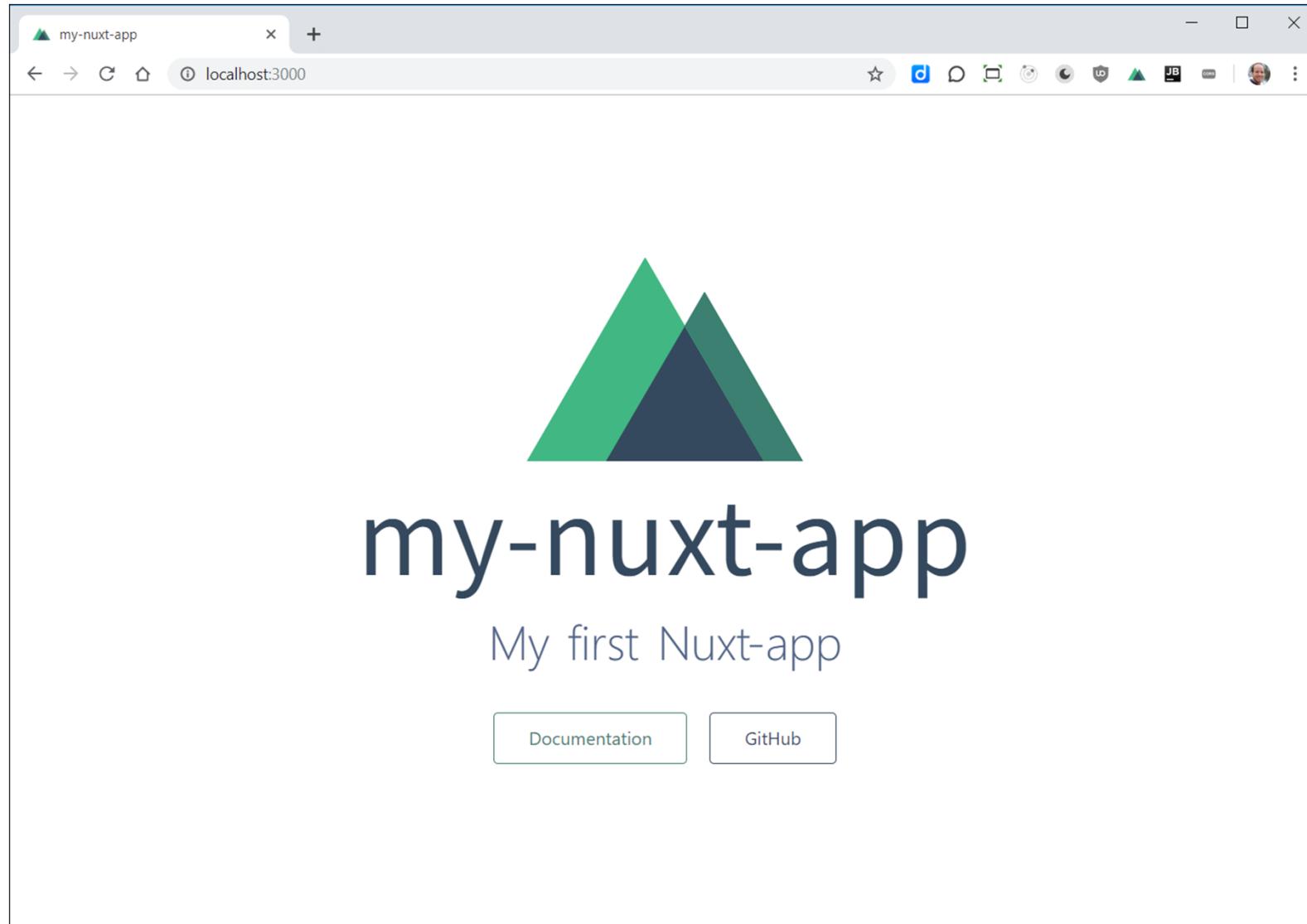
i Preparing project for development
i Initial build may take a while
✓ Builder initialized
✓ Nuxt files generated

* Client [██████████] building (28%) 157/167 modules 10 active
  babel-loader » vue-loader » .nuxt\components\nuxt-loading.vue

* Server [██] building (13%) 32/36 modules 4 active
  babel-loader » vue-loader » .nuxt\components\nuxt-loading.vue

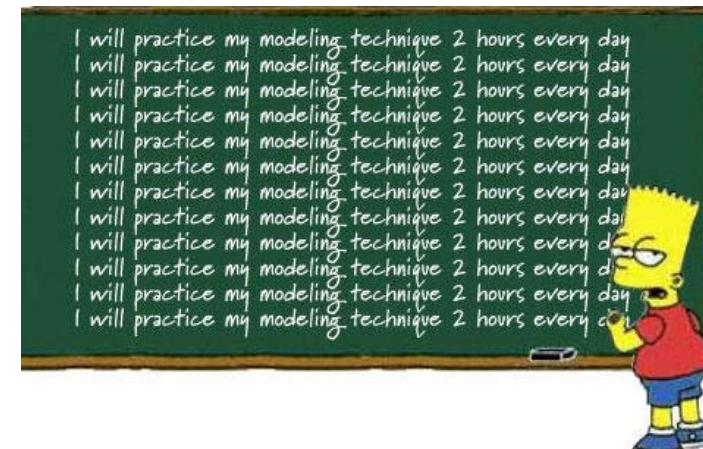
10:11:06
10:11:06
10:11:06
10:11:06
```

Default address: localhost:3000



Workshop

- Go ahead and create your first Nuxt project
 - Use `npx create-nuxt-app` or `yarn create nuxt-app`
- Make your own choices in the cli prompts
- Run the new application



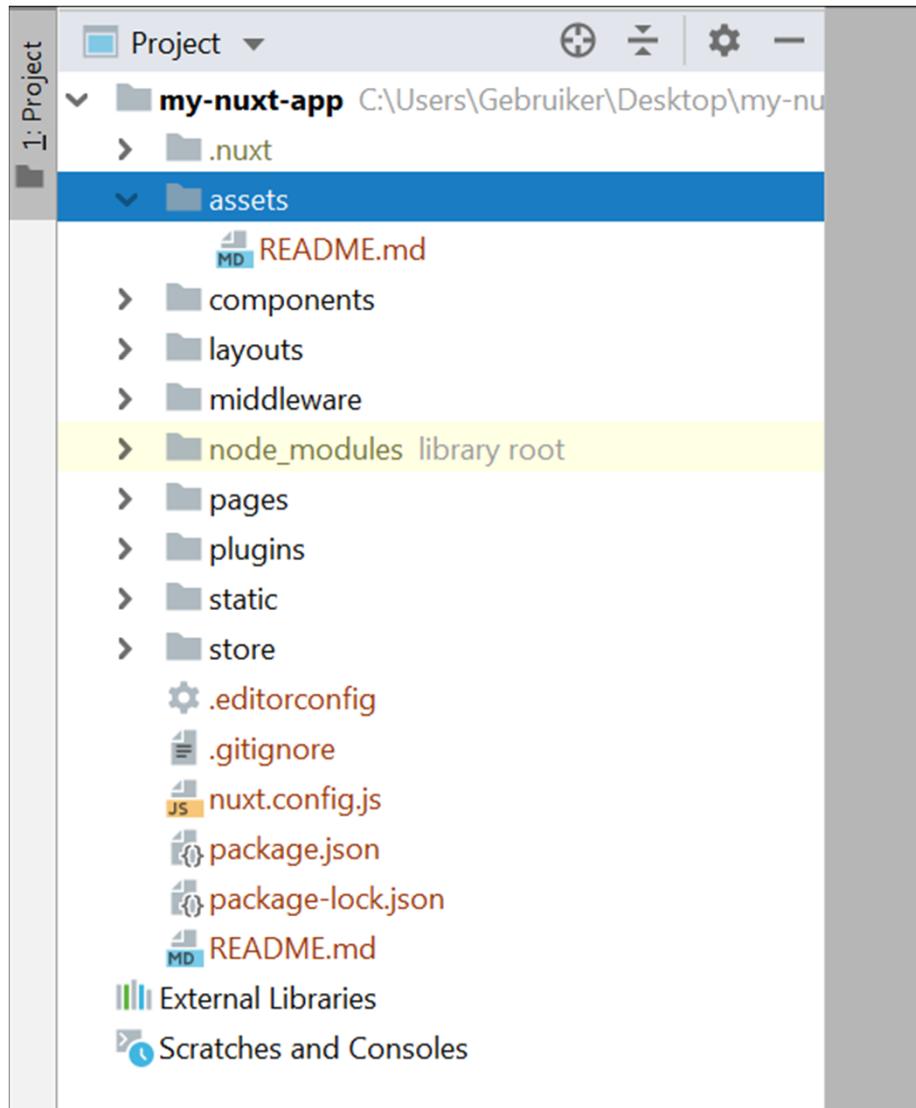


A look at the project structure

Nuxt has generated a lot of default folders for you



Default folder structure

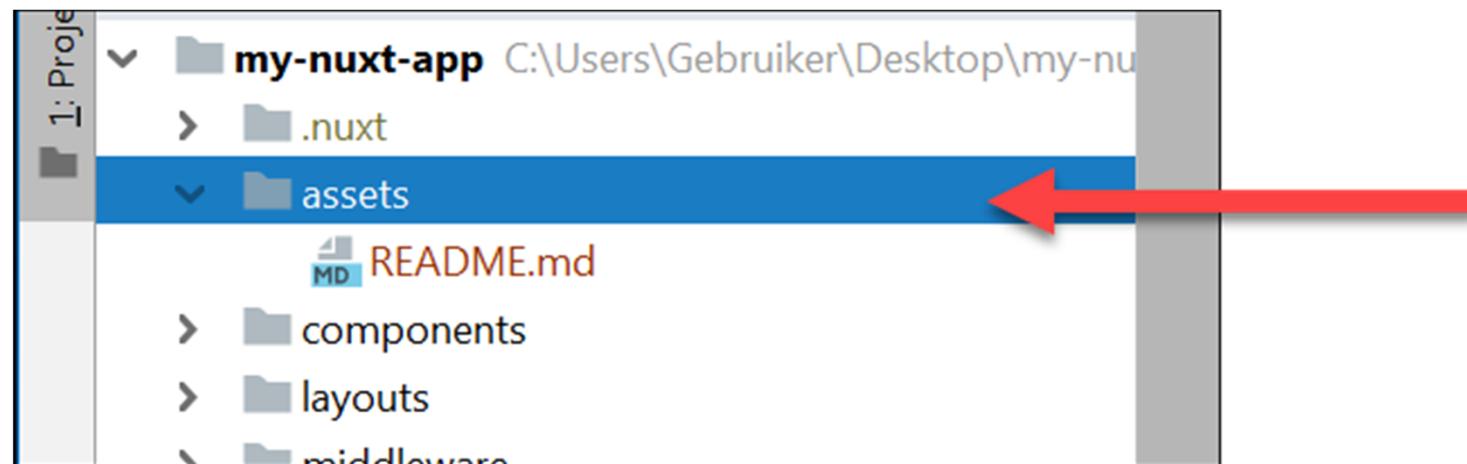


A lot of default folders and files

Assets directory



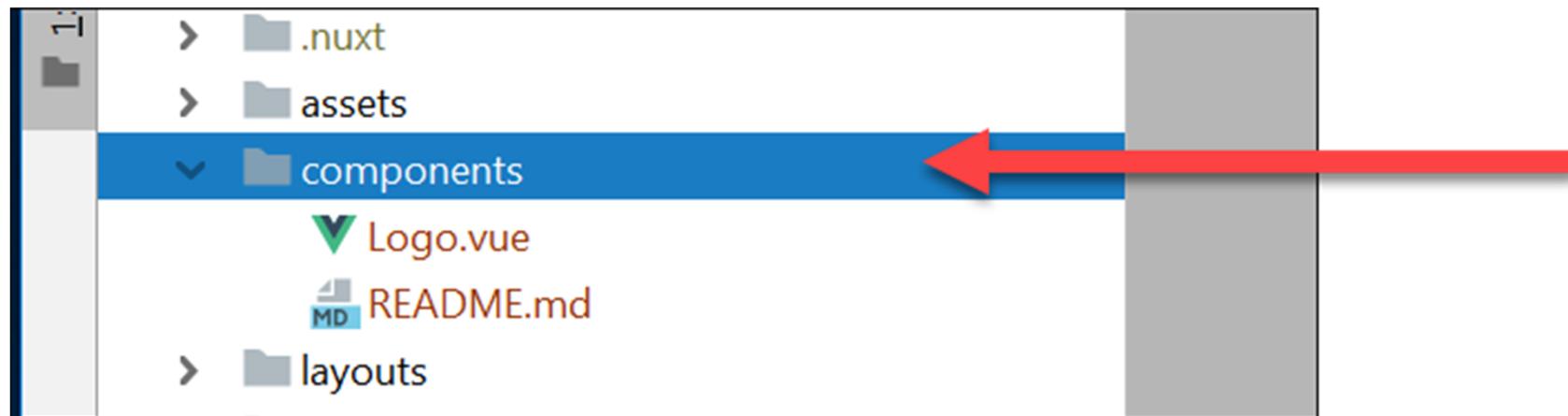
- Contains static files that are used in the application
 - Sass/Less files
 - Images
 - Fonts
- The same directory as a standard Vue-cli project





Components directory

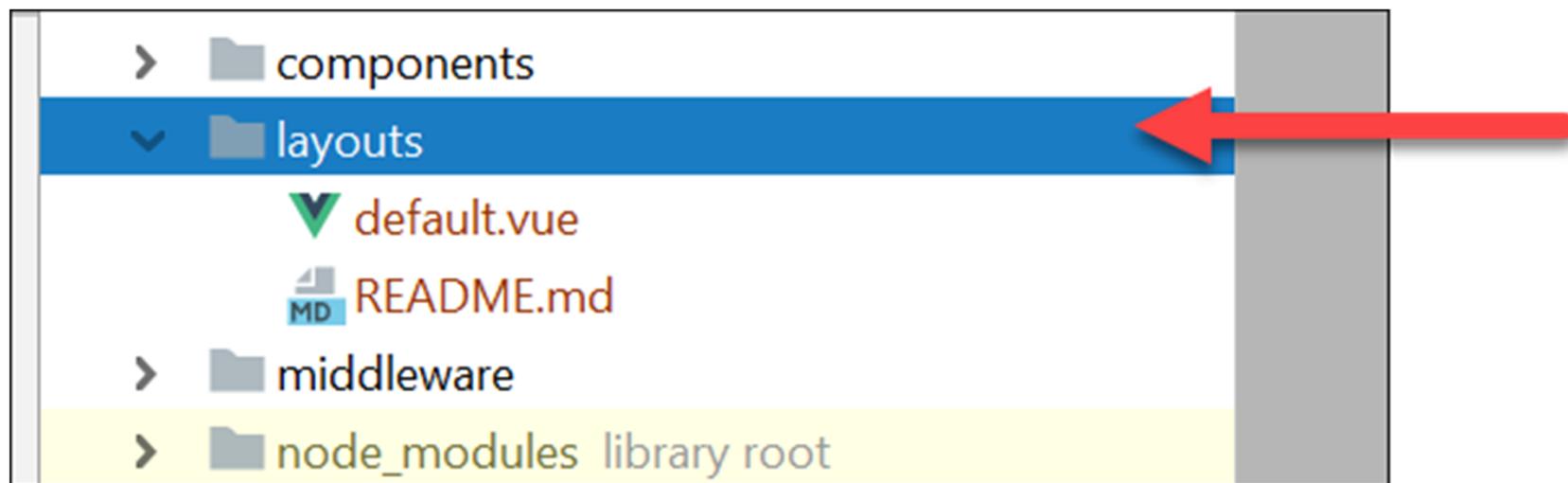
- Contains your Single File Components
 - .vue files
- The same directory as a standard Vue-cli project





Layouts directory

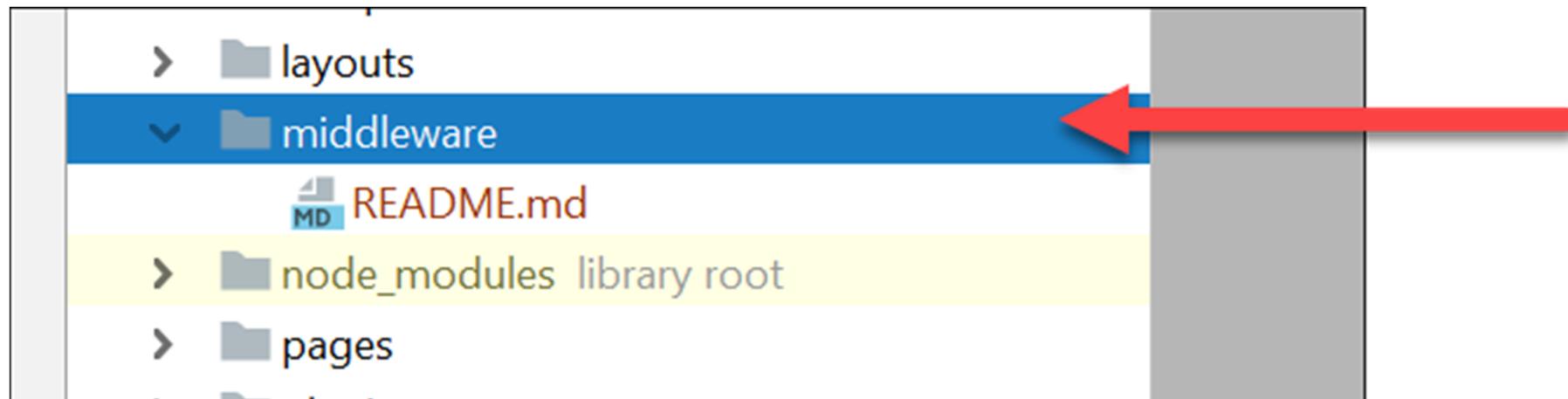
- Contains layouts for your application
 - Layouts are like the 'master pages' of your application
 - For instance: a lay-out for the homepage, a lay-out for detail pages, a lay-out with/without sidebar navigation, and so on.
- This is specific to a nuxt-project



Middleware directory



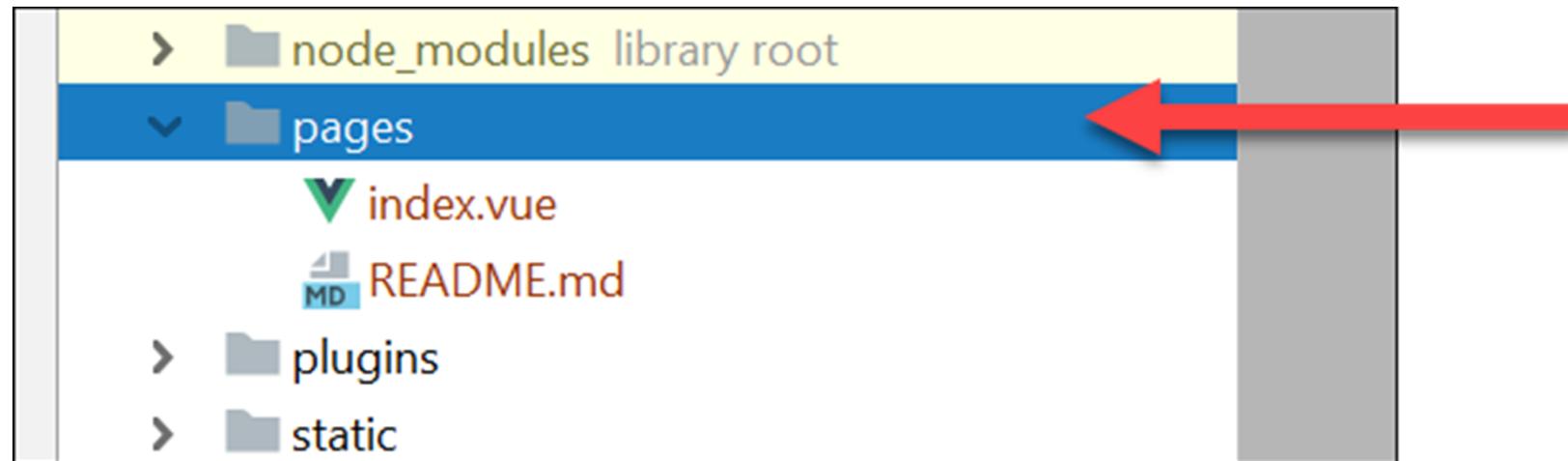
- Contains middleware functions
 - Middleware are custom functions that can be run before rendering a page, or group of pages.
- This is specific to a nuxt-project
- You can delete it if you don't use middleware (yet)



Pages directory



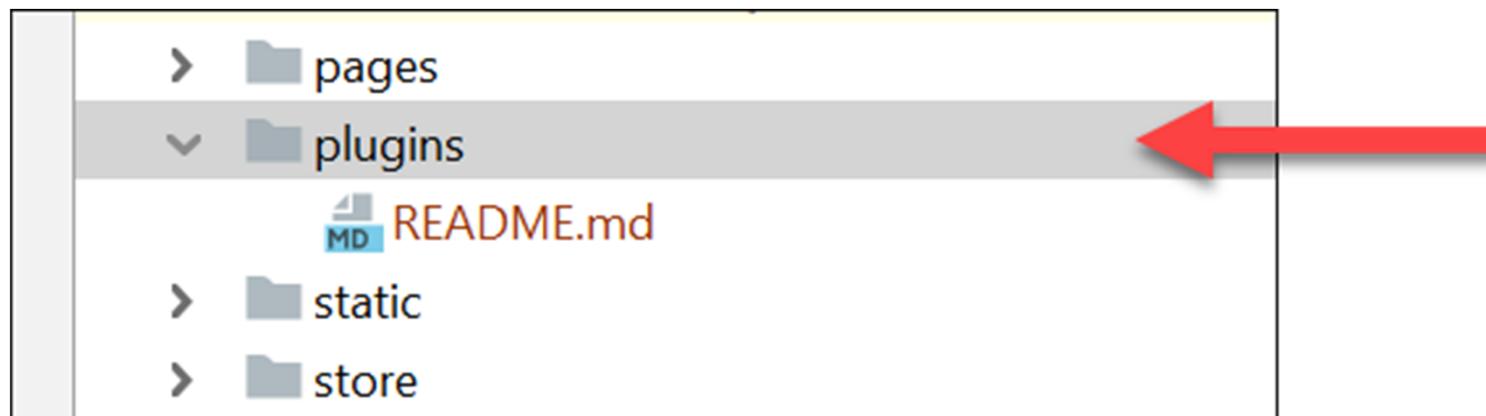
- Contains the Application Views and Routes
 - Nuxt reads all the pages in this directory and creates the routes,
 - based upon the pages and structure of this directory.
 - Pages are also .vue-files
- 'Pages' can contain multiple components



Plugins directory



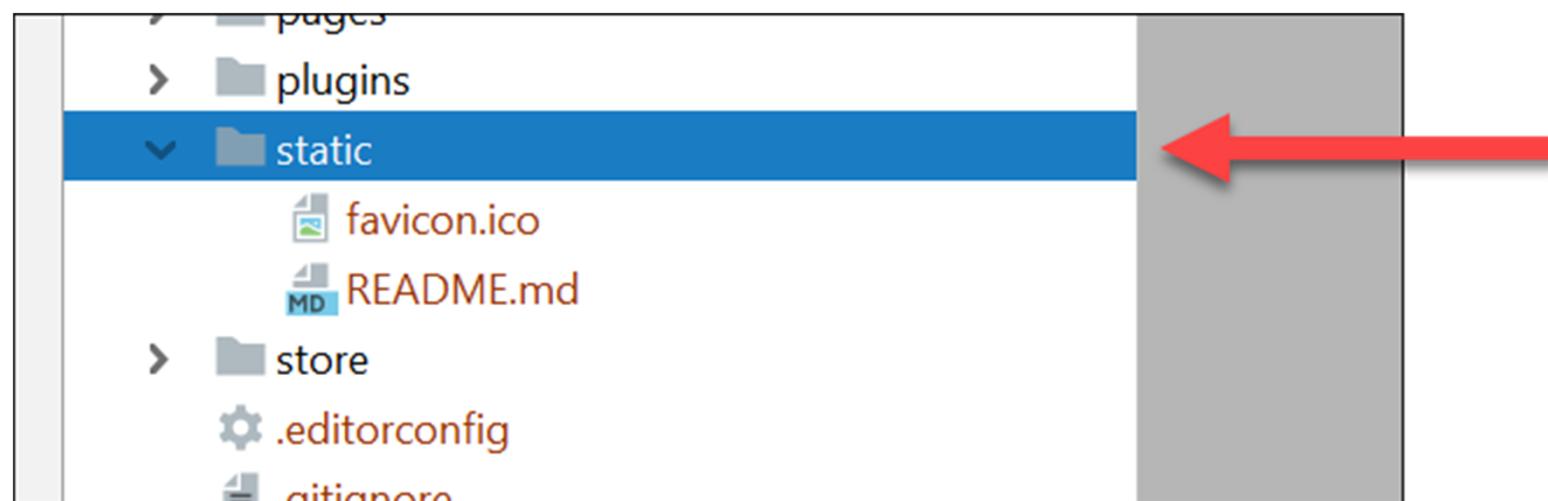
- Contains JavaScript plug-ins that run *before* the root Vue application is mounted
 - General code that you want to run before the app starts.
 - For instance, set up keys on API's, custom modules, load libraries and so on
- You can delete the directory if you don't use plug-ins (yet)



Static directory



- All files in this directory are mapped to the root (/) of the application
 - Like logo's, robots.txt, favicon and so on.
 - Downloadable files (.zip, .mp3, .docx, etc)
- You can delete the directory if you don't use static files (yet)
- You *should* delete the readme-file in production



Store directory

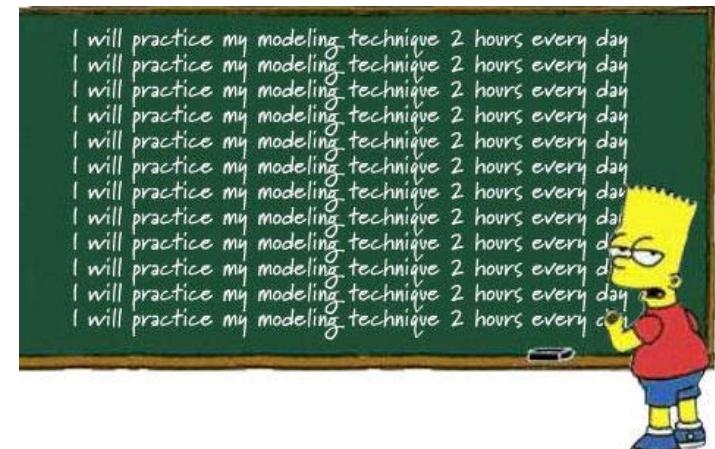


- Contains the vuex store
 - Vuex is available in your application, but disabled by default
 - Store is activated as soon as you put JavaScript-files in this directory
- You can delete the directory if you don't use a store (yet)



Workshop

- Look at the folder structure of your created application and make sure you understand the purposes of every folder
- Read the README.md-files in every folder
- Optional: delete the folders that you are sure you're not going to use





Customizing pages

Edit the default homepage and other pages in your nuxt project

Customizing the home page



- Update /pages/index.vue, as usual
- Delete unwanted HTML
- Delete unwanted CSS
- Delete/update script

A screenshot of a code editor showing two files: index.vue and Logo.vue. The index.vue file contains Vue.js template and script blocks. The template block defines a container div containing a logo component and two heading elements (h1 and h2). The script block imports the Logo component from a local file and defines the export default object with components: { Logo }.

```
index.vue x Logo.vue x
1 <template>
2   <div class="container">
3     <Logo/>
4     <h1 class="alert alert-danger">
5       | How to learn C#
6     </h1>
7     <h2 class="subtitle">
8       | My first Nuxt-app
9     </h2>
10    </div>
11  </template>
12
13  <script>
14    import Logo from '~/components/Logo.vue'
15
16    export default {
17      components: {
18        Logo
19      }
20    }
21  </script>
```

Nuxt Page Components



- Pages are like the 'Smart Components' in your app
 - They are .vue files in the /pages directory
 - Nuxt will automatically create routes for every page found here
 - Pages hold your logic and data (if not inside a store)
- Components are like the 'View components'
 - Components can be used on multiple pages
 - Components typically don't do much logic.
 - They communicate with the pages via props and events.
- We can directly navigate to a route via the URL

Nuxt Pages



The screenshot shows a code editor with a file tree on the left and a code editor window on the right.

File Tree:

- > node_modules library root
- <-- pages (highlighted by a red arrow)
- <-- history.vue
- <-- index.vue
- MD README.md
- > plugins
- > static

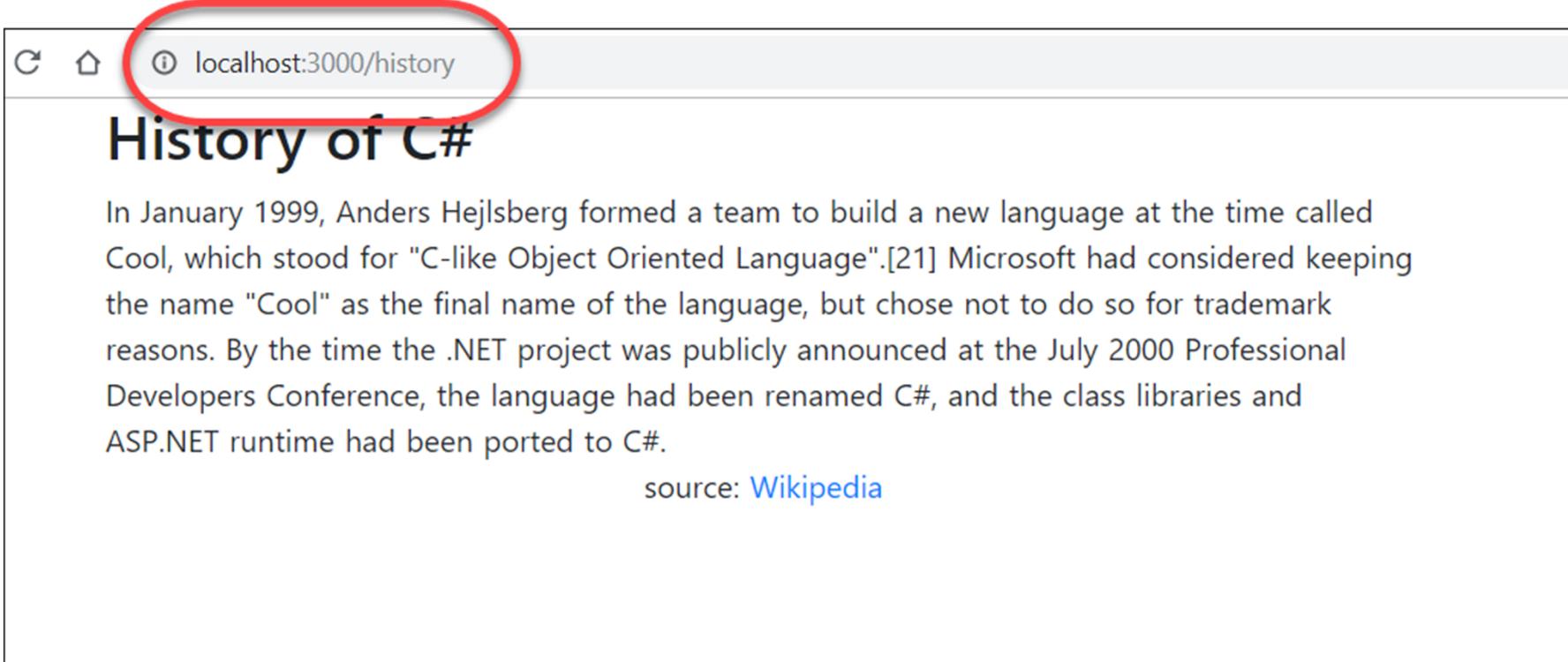
Code Editor:

```
1 <template>
2   <div class="container">
3     <h2>History of C#</h2>
4     <div>
5       In January 1999, Anders Hejlsberg formed a team to build a
6       new language at the time called Cool, which stood for "C-like Object O
7       considered keeping the name "Cool" as the final name of the language,
8       reasons. By the time the .NET project was publicly announced at the Ju
9       the language had been renamed C#, and the class libraries and ASP.NET
10      </div>
11      <div class="text-sm-center">
12        source: <a href="https://en.wikipedia.org/wiki/C_Sharp_(programming_la
13      </div>
14    </div>
15  </template>
16
17  <script>
18    export default {
19      name: "history"
20    }
21  </script>
```

The code editor shows a template section with an h2 header and a nested div containing text about the history of C#. The script section defines a default export named "history". A yellow highlight covers the entire template and script sections. A small orange dot is located on the line starting with "</div>" in the template section.



Navigate via URL



The screenshot shows a web browser window with a red circle highlighting the URL bar. The URL displayed is `localhost:3000/history`. Below the URL bar, the page title is **History of C#**. The main content area contains a paragraph about the history of C#. The source of the information is cited as [Wikipedia](#).

In January 1999, Anders Hejlsberg formed a team to build a new language at the time called Cool, which stood for "C-like Object Oriented Language".[21] Microsoft had considered keeping the name "Cool" as the final name of the language, but chose not to do so for trademark reasons. By the time the .NET project was publicly announced at the July 2000 Professional Developers Conference, the language had been renamed C#, and the class libraries and ASP.NET runtime had been ported to C#.

source: [Wikipedia](#)

Where is my App.vue?



- In standard Vue applications, the top level component is `App.vue`
- In Nuxt applications this is `../layouts/default.vue`
- So, global styles in the lay-out are applied to all pages that use it

The screenshot shows a code editor with the project structure of a Nuxt application on the left and the content of the `default.vue` file on the right.

Project Structure:

- `my-nuxt-app` (Project)
- `my-nuxt-app` (Folder)
 - `.nuxt`
 - `assets`
 - `components`
 - `layouts` (Folder)
 - `default.vue` (Selected, highlighted with a red arrow)
 - `index.vue`
 - `history.vue`
 - `README.md`
 - `middleware`
 - `node_modules` (library root)
 - `pages` (Folder)
 - `history.vue`
 - `index.vue`
 - `README.md`
 - `plugins`
 - `static`
 - `store`

Code Editor Content:

```
<template>
  <div>
    <nuxt/>
  </div>
</template>

<style>
  /* h2 now has a global style */
  h2 {
    color: cornflowerblue;
  }
</style>
```

Browser like...

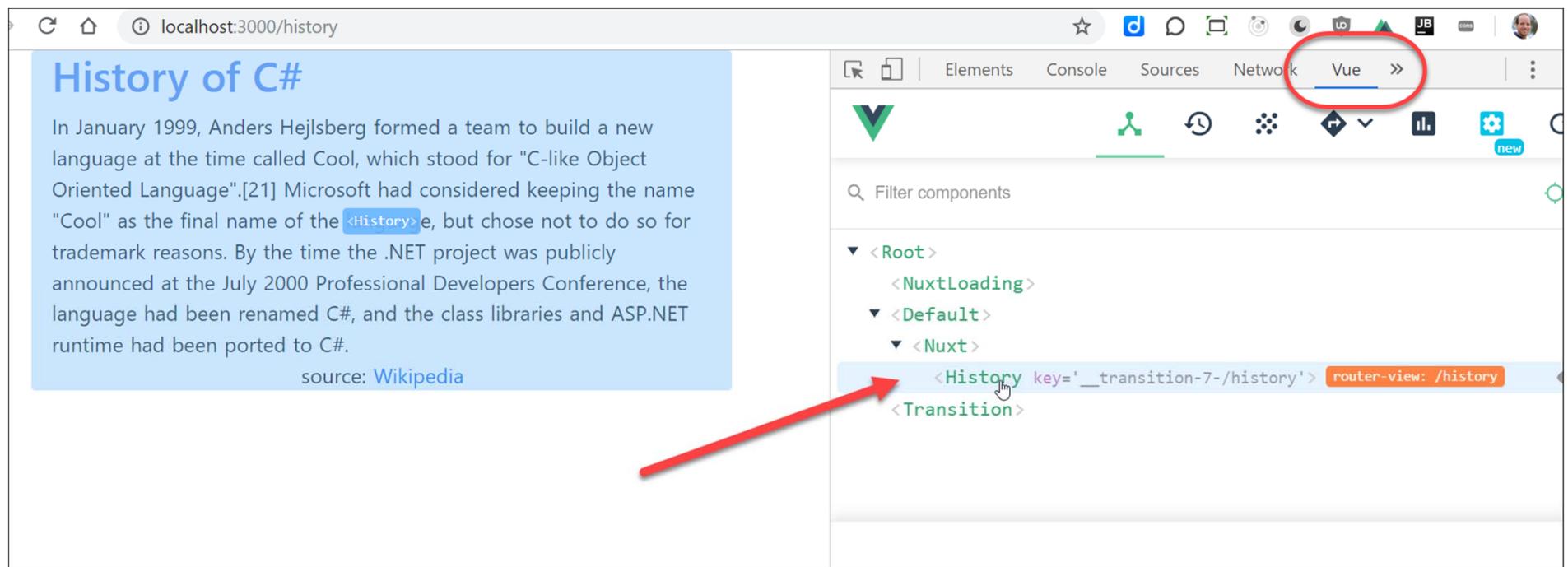


localhost:3000/history

History of C#

In January 1999, Anders Hejlsberg formed a team to build a new language at the time called Cool, which stood for "C-like Object Oriented Language".[21] Microsoft had considered keeping the name "Cool" as the final name of the language, but chose not to do so for trademark reasons. By the time the .NET project was publicly announced at the July 2000 Professional Developers Conference, the language had been renamed C#, and the class libraries and ASP.NET runtime had been ported to C#.

source: [Wikipedia](#)



Use the Vue DevTools to see the composition and structure of the generated application

Code splitting by default

How to learn C#

My first Nuxt-app

Just index page loaded

History of C#

In January 1999, Anders Hejlsberg formed a team to build a new language at the time called Cool, which stood for "C-like Object Oriented Language".[21] Microsoft had considered keeping the name "Cool" as the final name of the language, but chose not to do so for trademark reasons. By the time the .NET project was publicly announced at the July 2000 Professional Developers Conference, the language had been renamed C#, and the class libraries and ASP.NET runtime had been ported to C#.

source: [Wikipedia](#)

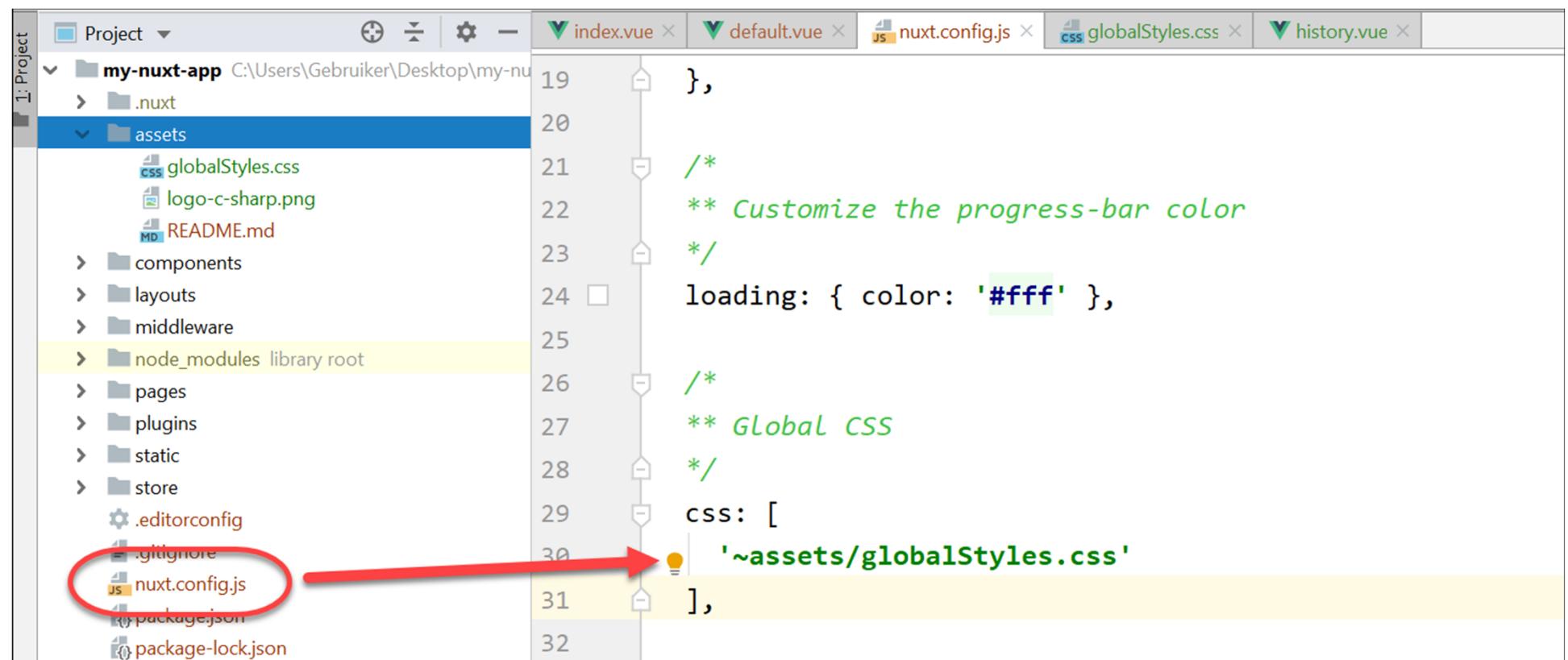
Just history page loaded

Resource	Status	Type	Name	Size	Time
app.js	200	script	(index)	33...	3 ms
vendors.app.js	200	script	(index)	619...	44 ...
app.js	200	script	(index)	1.8 ...	84 ...
app.b437138c50...	200	script	(index)	175...	87 ...
pages_index.js	200	script	(index)	3.0 ...	88 ...
backend.js	200	script	VM460:5	390...	24 ...
logo-c-sharp.png	200	png	(index)	20...	49 ...
client	200	eve...	(index)	63...	37 ...
ng-validate.js	200	script	content...	55...	65 ...
icon16.png	200	image	(index)	200 B	2 ms
FN-close.png	200	image	(index)	12 B	1 ms
tab-logo.png	200	image	(index)	39 B	1 ms
inject.js	200	script	content...	1.5 ...	8 ms

Resource	Status	Type	Name	Size	Time
app.js	200	script	history	1.8 ...	84 ...
vendors.app.js	200	script	history	619...	42 ...
app.js	200	script	history	175...	86 ...
app.b437138c50...	200	script	history	3.0 ...	87 ...
pages_history.js	200	script	history	9.4 ...	87 ...
backend.js	200	script	VM500:5	390...	23 ...
client	200	eve...	history	63...	2.52 s
ng-validate.js	200	script	content...	55...	69 ...
icon16.png	200	png	jquery-1...	404 B	39 ...
FN-close.png	200	png	jquery-1...	927 B	12 ...
tab-logo.png	200	png	jquery-1...	1.2 ...	18 ...
inject.js	200	script	content...	1.5 ...	8 ms

Creating global CSS

- Expand nuxt.config.js to apply global style rules
- Create a global rule file in /assets, for instance like:



The screenshot shows a code editor interface with the following details:

- Project Structure:** The left sidebar shows a tree view of the project. The "assets" folder under "my-nuxt-app" is selected and highlighted in blue. Other visible files include ".nuxt", "index.vue", "default.vue", "nuxt.config.js", "globalStyles.css", "history.vue", "logo-c-sharp.png", and "README.md".
- nuxt.config.js Content:** The right pane displays the contents of the nuxt.config.js file. A red arrow points from the "nuxt.config.js" entry in the project tree to the line of code in the editor.
- Code Content:**

```
19     },
20
21     /*
22      ** Customize the progress-bar color
23     */
24     loading: { color: '#fff' },
25
26     /*
27      ** Global CSS
28     */
29     css: [
30       '~assets/globalStyles.css'
31     ],
32   }
```

```
my-nuxt-app assets globalStyles.css
Project my-nuxt-app C:\Users\Gebruiker\Desktop\my-nu
> .nuxt
> assets
  globalStyles.css
  logo-c-sharp.png
  README.md
> components
> layouts
> middleware
```

1: Project

1 p, div{
2 color: saddlebrown;
3 }
4

- The `~assets/...` is a pointer to the Nuxt root directory
and then the assets folder
- You can have an array of stylesheets (not just a single file)
- Also, you can point to external packages/libraries
- You can use .scss files. Nuxt will automatically guess the file type by its extension

More info: nuxtjs.org/api/configuration-css

The screenshot shows the Nuxt.js API documentation for the configuration-css section. The left sidebar lists various configuration options, with 'css' highlighted by a green bar. The main content area has a title 'The css Property' and a description explaining that Nuxt.js lets you define CSS files/modules/libraries globally. It includes a note about installing node-sass and sass-loader, followed by a code example and its type definition. A red arrow points from the 'npm install' command in the note to the code example below.

Nuxt.js lets you define the CSS files/modules/libraries you want to set globally (included in every page).

In case you want to use `sass` make sure that you have installed `node-sass` and `sass-loader` packages. If you didn't just

```
npm install --save-dev node-sass sass-loader
```

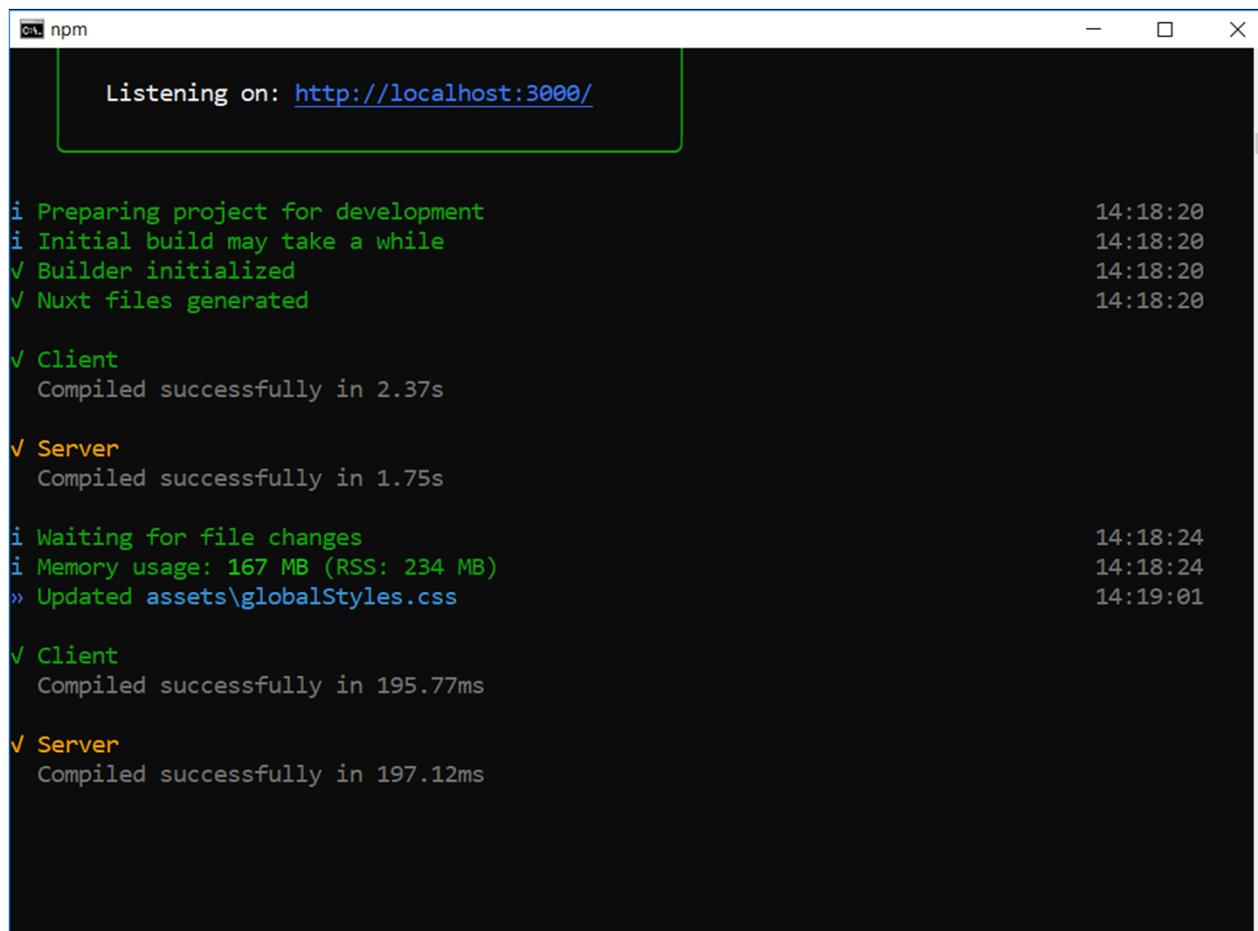
- Type: `Array`
 - Items: `string`

In `nuxt.config.js`, add the CSS resources:

```
export default {  
  css: [  
    // Load a Node.js module directly (here it's a Sass file)  
    'bulma',  
    // CSS file in the project  
    '@/assets/css/main.css',  
    // SCSS file in the project  
    '@/assets/css/main.scss'  
  ]  
}
```

NOTE! Restart dev server

- If you changed `nuxt.config.js`, make sure to restart the server
- The config file is only read on startup.



```
npm
Listening on: http://localhost:3000/

i Preparing project for development
i Initial build may take a while
v Builder initialized
v Nuxt files generated

v Client
Compiled successfully in 2.37s

v Server
Compiled successfully in 1.75s

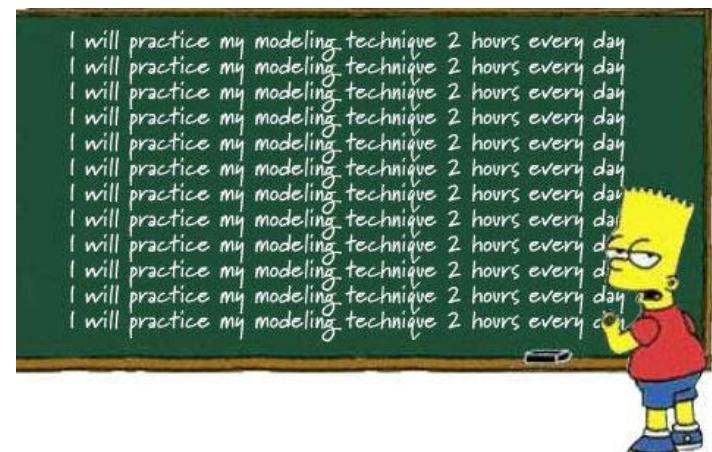
i Waiting for file changes
i Memory usage: 167 MB (RSS: 234 MB)
» Updated assets\globalStyles.css

v Client
Compiled successfully in 195.77ms

v Server
Compiled successfully in 197.12ms
```

Workshop

- Customize the home page of your app
- Create at least 2-3 new pages (products, about us, contact, etc)
- Create a global style and add it to nuxt.config.js.





Creating navigation

Global Navigation Bars and/or Main Menu for your site

Creating global navigation



- Drop global navigation in a layout file.
- Best practices:
 - Create a separate component for navigation, to keep files clean and simple
 - Name a global component something like `TheNavigation.vue` or `TheSidebar.vue`
 - <https://vuejs.org/v2/style-guide/#Single-instance-component-names-strongly-recommended>

Components that should only ever have a single active instance should begin with the `The` prefix, to denote that there can be only one.

This does not mean the component is only used in a single page, but it will only be used once *per page*. These components never accept any props, since they are specific to your app, not their context within your app. If you find the need to add props, it's a good indication that this is actually a reusable component that is only used once per page *for now*.

Bad

```
components/
|- Heading.vue
|- MySidebar.vue
```

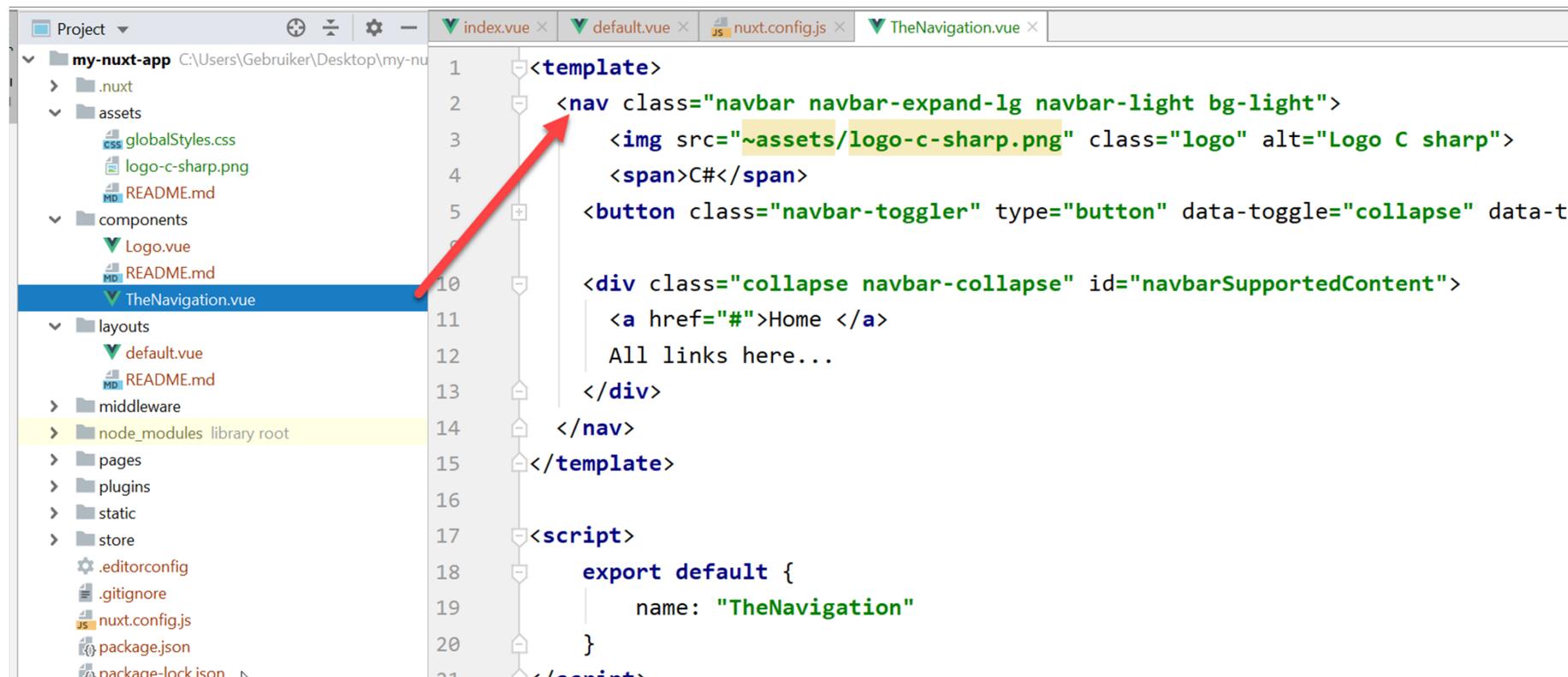
Good

```
components/
|- TheHeading.vue
|- TheSidebar.vue
```

TheNavigation.vue



- Contains – in this example – a default Bootstrap Navigation Bar



```
Project my-nuxt-app C:\Users\Gebruiker\Desktop\my-nuxt
  .nuxt
  assets
    globalStyles.css
    logo-c-sharp.png
    README.md
  components
    Logo.vue
    README.md
  TheNavigation.vue
  layouts
    default.vue
    README.md
  middleware
  node_modules library root
  pages
  plugins
  static
  store
    .editorconfig
    .gitignore
    nuxt.config.js
    package.json
    package-lock.json

<template>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    
    <span>C#</span>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-t
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <a href="#">Home </a>
      All links here...
    </div>
  </nav>
</template>
<script>
  export default {
    name: "TheNavigation"
</script>
```

Add it to ../layouts/default.vue



- Add your navigation before the `<nuxt />` element
- All pages are projected inside `<nuxt />`

```
my-nuxt-app > layouts > default.vue
Project . . . . .
my-nuxt-app C:\Users\Gebruiker\Desktop\my-nu
> .nuxt
<-- assets
  globalStyles.css
  logo-c-sharp.png
  README.md
<-- components
  Logo.vue
  README.md
  TheNavigation.vue
<-- layouts
  default.vue
  README.md
> middleware
> node_modules library root

<template>
  <div>
    <TheNavigation />
    <nuxt/>
  </div>
</template>
<style>
  /* h2 now has a global style */
  h2 {
    color: cornflowerblue;
  }
</style>
```

Browser



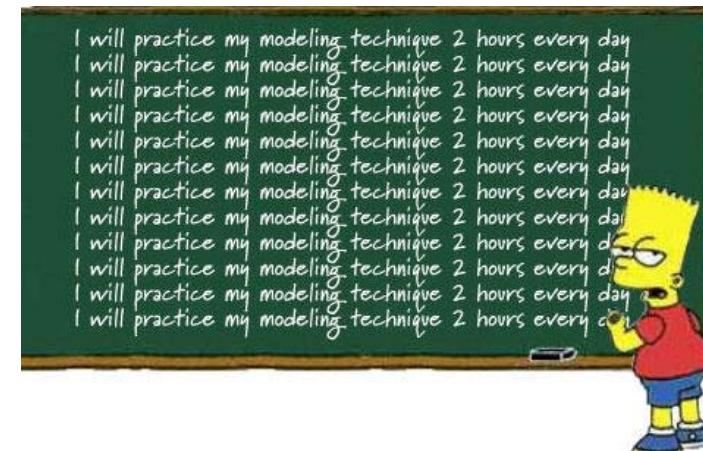
- Links are activated/filled in momentarily...

The image displays two side-by-side browser windows. Both windows have a header bar with back, forward, refresh, and home icons, and a URL field showing 'localhost:3000'.
The left window shows the homepage of a C# learning application. It features a blue hexagonal logo with a white 'C#' symbol. Below it, a teal banner contains the text 'How to learn' in white. Underneath the banner, the text 'My first Nuxt-app' is displayed in blue. The URL in the address bar is 'localhost:3000'.
The right window shows a sub-page titled 'History of C#' in blue. The page content discusses the history of the C# language, mentioning Anders Hejlsberg's team formation in 1999, Microsoft's consideration of 'Cool' as the name, and the public announcement at the July 2000 Professional Developers Conference. It also notes that the ASP.NET runtime had been ported to C#. The URL in the address bar is 'localhost:3000/history'. A red oval highlights the URL bar of the right window.

source: [Wiki](#)

Workshop

- Create a Main Navigation system for your app
- It can be a main menu header, sidebar or whatever.
- Be sure to create a component of it, and use it in layouts/default.vue





Linking between pages

Using the `<nuxt-link>` element

Linking between pages



- In nuxt, you create links between pages using the `<nuxt-link>` element
- It looks like the standard `<router-link>` of Vue.
- If you use regular `` tags, the page will refresh



Using <nuxt-link>

```
index.vue x default.vue x nuxt.config.js x TheNavigation.vue x
1 <template>
2   <nav class="navbar navbar-expand-lg navbar-light bg-light">
3     
4     <span>C#</span>
5     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
6       <span>≡</span>
7     </button>
8     <div class="collapse navbar-collapse" id="navbarSupportedContent">
9       <ul class="navbar-nav mr-auto">
10      <li class="nav-item">
11        <nuxt-link exact class="nav-link" to="/">Home</nuxt-link>
12      </li>
13      <li class="nav-item">
14        <nuxt-link exact class="nav-link" to="/history">History</nuxt-link>
15      </li>
16    </ul>
17  </div>
18 </nav>
19 </div>
```



Browser

localhost:3000

C# Home History

localhost:3000/history

C# Home History

How to I

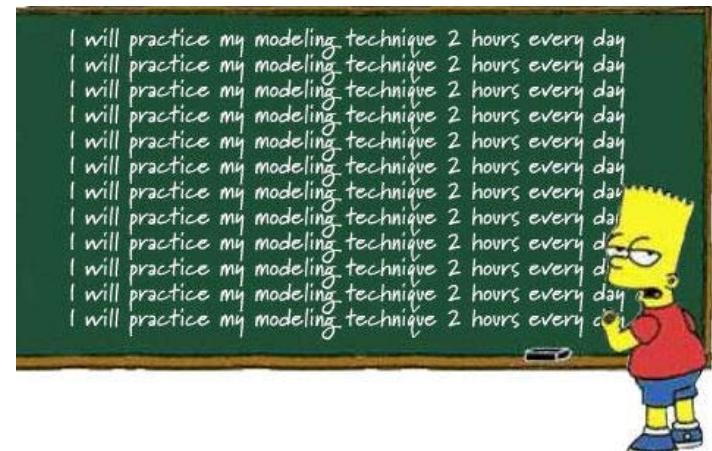
My first Nux

History of C#

In January 1999, Anders Hejlsberg formed a team to build a new lang
Microsoft had considered keeping the name "Cool" as the final name
project was publicly announced at the July 2000 Professional Develop
ASP.NET runtime had been ported to C#.

Workshop

- Add links to the main navigation of your app
- Test in the browser if they work



Dynamic Routing



*You create a dynamic route by using
the underscore prefix*

- First, create a component <PostList/>, containing an overview of some dummy posts
- <https://jsonplaceholder.typicode.com/posts>



Dynamic Routing

```
PostList.vue (highlighted by red circle):
1 <template>
2   <div class="container">
3     <h2>All Posts</h2>
4     <ul class="list-group">
5       <li v-for="post in posts" :key="post.id"
6           class="list-group-item">
7         <nuxt-link :to=`/posts/${post.id}`>
8           {{ post.title}}
9         </nuxt-link>
10        </li>
11      </ul>
12    </div>
13  </template>
```

```
default.vue (highlighted by yellow bar):
15 <script>
16   import posts from '../data/posts'
17
18   export default {
19     name: "PostList",
20     data() {
21       return {
22         posts
23       }
24     }
25   }
26 </script>
```

Dynamic Routing

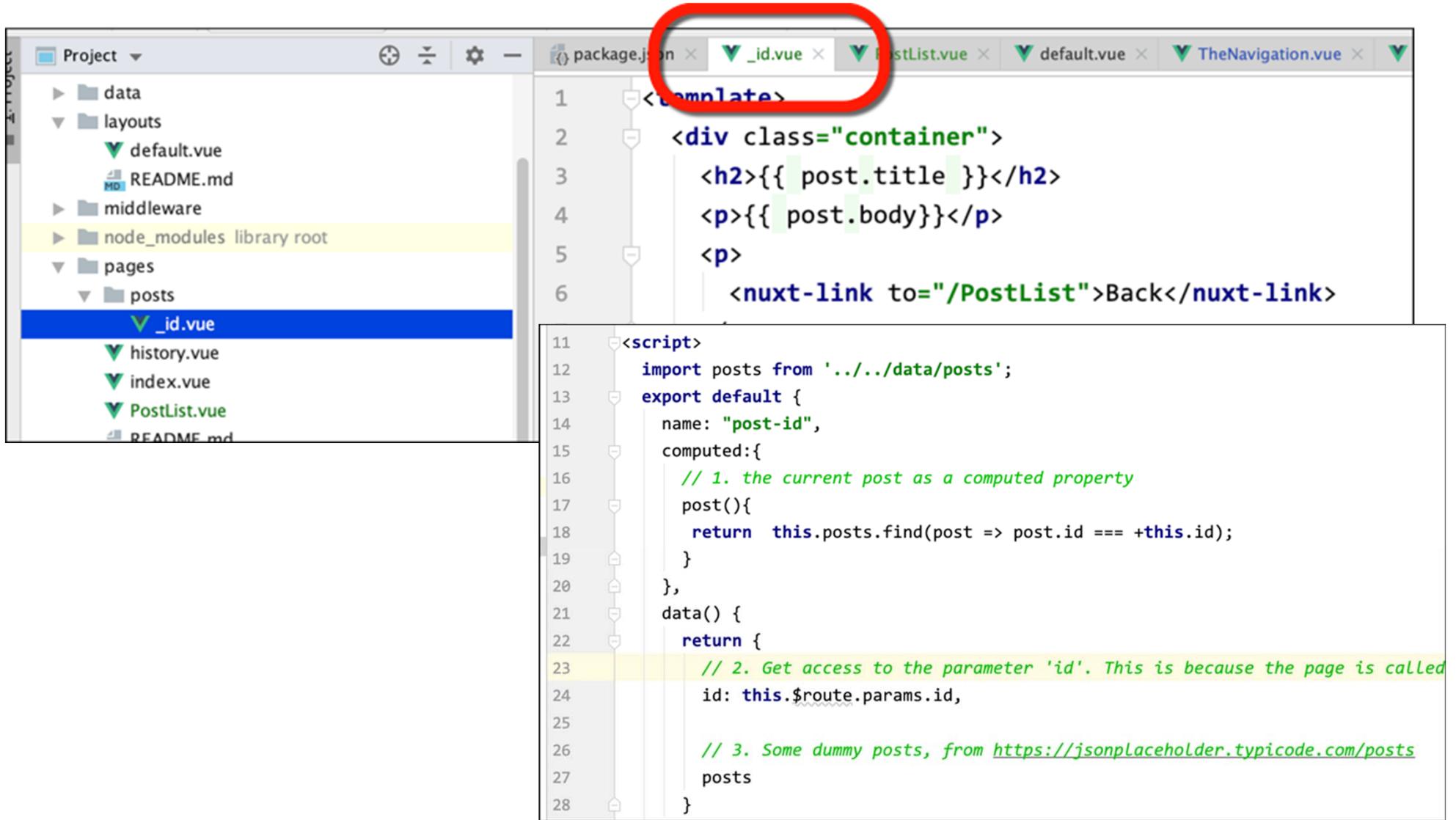
A screenshot of a web browser window. The address bar shows 'localhost:3000/PostList'. The page header includes a blue hexagonal icon with a white 'C#' and 'Home', 'History', 'Post List' links. The main content area has a light gray background with horizontal lines separating sections. The title 'All Posts' is in blue. Below it is a list of seven items, each starting with a blue placeholder text:

- sunt aut facere repellat provident occaecati excepturi optio reprehenderit
- qui est esse
- ea molestias quasi exercitationem repellat qui ipsa sit aut
- eum et est occaecati
- nesciunt quas odio
- dolorem eum magni eos aperiam quia
- magnam facilis autem

Dynamic Routing



- Now, create a directory with a dynamic file



The screenshot shows a code editor with a sidebar displaying a project structure:

- Project
- data
- layouts
- default.vue
- README.md
- middleware
- node_modules library root
- pages
- posts
- _id.vue
- history.vue
- index.vue
- PostList.vue
- README.md

The _id.vue file is selected in the sidebar and is currently being edited. The code editor has tabs for package.json, _id.vue, PostList.vue, default.vue, and TheNavigation.vue. The _id.vue tab is active.

The code in the template section is:

```
<template>
  <div class="container">
    <h2>{{ post.title }}</h2>
    <p>{{ post.body }}</p>
    <p>
      <nuxt-link to="/PostList">Back</nuxt-link>
    </p>
  </div>
```

The script section contains the following code:

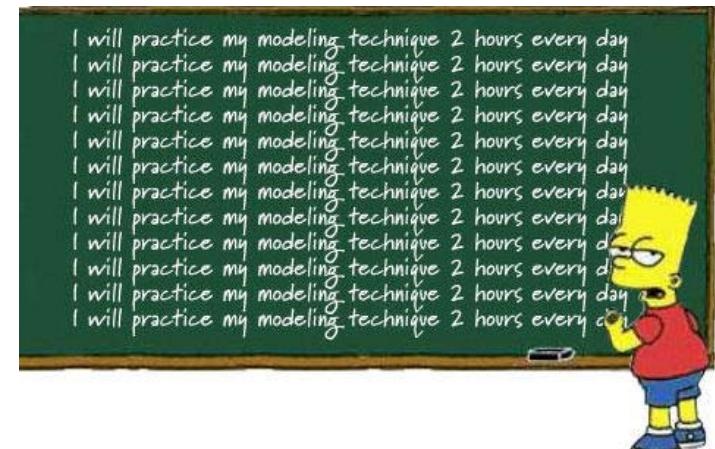
```
<script>
  import posts from '../data/posts';
  export default {
    name: "post-id",
    computed: {
      // 1. the current post as a computed property
      post(){
        return this.posts.find(post => post.id === +this.id);
      }
    },
    data() {
      return {
        // 2. Get access to the parameter 'id'. This is because the page is called
        id: this.$route.params.id,
        // 3. Some dummy posts, from https://jsonplaceholder.typicode.com/posts
        posts
      }
    }
  }
}
```

A red circle highlights the template section of the code. A yellow box highlights the data() block in the script section, with the following annotations:

- // 2. Get access to the parameter 'id'. This is because the page is called
- id: this.\$route.params.id,
- // 3. Some dummy posts, from https://jsonplaceholder.typicode.com/posts

Workshop

- Add dynamic links to the Nuxt application
- You can use the address jsonplaceholder.typicode.com/posts for some random data / fake blog posts
- Test in the browser if they work
- Example `.../800-nuxt`





SEO and meta tags

Using the built-in `vue-meta` options

SEO and Page Meta Tags



- Nuxt is great for Server Side Rendering
 - Pages are already processed on the server
- Create SEO and Social Media Tags by setting the `head` property of components/pages in the project
 - This is actually an override of the `metaInfo: { ... }` property found in `vue-meta`
- Head is a function that returns an object with the content.
 - ```
head() {
 return {
 title: 'The Home Page...'
 }
}
```

# Other tags



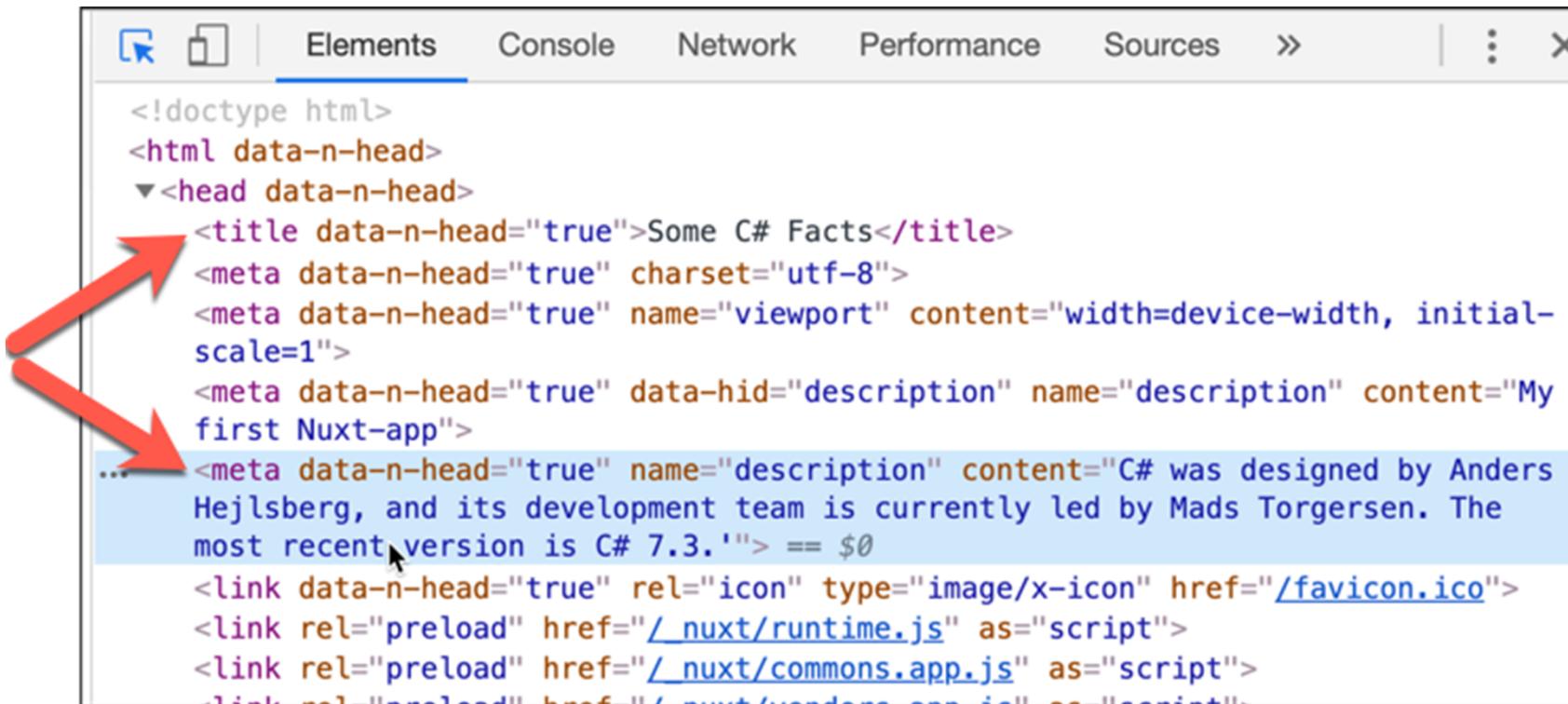
- `description` is used for the description of the page.
- You can access the `this` keyword for underlying data properties of the page
- Other meta tags: use an array
  - `Meta [`
    - `{ name: 'charset', content: 'utf-8' },`
    - `{ name: 'twitter:title', content: '...' },`
    - `{ ...other social media and meta tags... }`
  - `]`
- Inspect the `<head>` of the document in the browser DevTools to see the meta tags appear



```
package.json × _id.vue × PostList.vue × default.vue × TheNavigation.vue × seo.vue × index.vue × posts.js ×
15 data() {
16 return {
17 title: 'Some C# Facts'
18 }
19 },
20 head() {
21 return {
22 title: this.title,
23 meta: [
24 {
25 name: 'description',
26 content: 'C# was designed by Anders Hejlsberg, and its development team'
27 }
28]
29 }
30 }

```

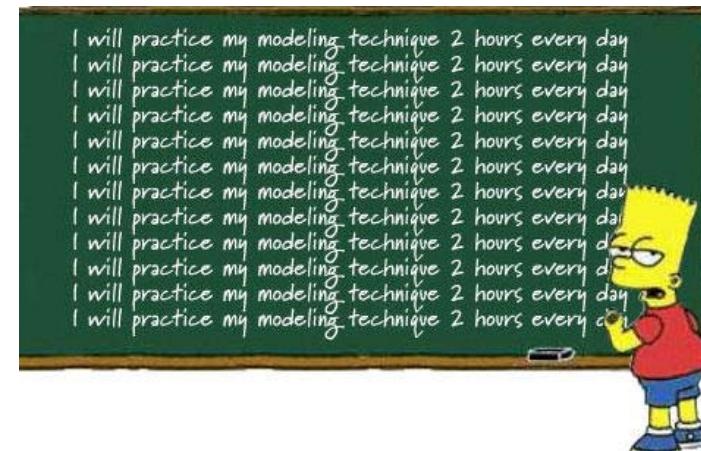
# In the DevTools



```
<!doctype html>
<html data-n-head=>
 <head data-n-head=>
 <title data-n-head="true">Some C# Facts</title>
 <meta data-n-head="true" charset="utf-8">
 <meta data-n-head="true" name="viewport" content="width=device-width, initial-scale=1">
 <meta data-n-head="true" data-hid="description" name="description" content="My first Nuxt-app">
 <meta data-n-head="true" name="description" content="C# was designed by Anders Hejlsberg, and its development team is currently led by Mads Torgersen. The most recent version is C# 7.3."> == $0
 <link data-n-head="true" rel="icon" type="image/x-icon" href="/favicon.ico">
 <link rel="preload" href="/_nuxt/runtime.js" as="script">
 <link rel="preload" href="/_nuxt/commons.app.js" as="script">
 <link rel="preload" href="/_nuxt/vendors.app.js" as="script">
```

# Workshop

- Add SEO tags to your pages
- Use basic title and description tags
- Add meta tags for Facebook, Instagram and/or Twitter
  - Look up for yourself which key/value-pairs they use
- Check if they appear in the browser





# Build and deploy

Sending your nuxt-application to the outside world

# Build & deploy Nuxt applications



- Build and serve your Nuxt.js App
- Local building:
  - `nuxt build && nuxt start` – build and test on local machine, with minified and uglified code.
  - `nuxt generate` – build and generate, great for SSR

COMMAND	DESCRIPTION
<code>nuxt</code>	Launch a development server on <code>localhost:3000</code> with hot-reloading.
<code>nuxt build</code>	Build your application with webpack and minify the JS & CSS (for production).
<code>nuxt start</code>	Start the server in production mode (after running <code>nuxt build</code> ).
<code>nuxt generate</code>	Build the application and generate every route as a HTML file (used for static hosting).

# npm run generate



The screenshot shows a terminal window with the following content:

```
Terminal: Terminal × +
> nuxt generate ← Red arrow pointing to this line

i Production build 20:54:17
✓ Builder initialized 20:54:17
✓ Nuxt files generated 20:54:17

• Client [progress bar] compiling (0%)

✓ Client
Compiled successfully in 14.94s

• Server [progress bar] additional chunk assets processing (91%)
```

The terminal window has a sidebar on the left with icons for Favorites, npm, and Structure. At the bottom, there are tabs for Dilbert, Version Control, Terminal, Run, TODO, and Event Log (which has 1 entry).



# Prerendering dynamic routes

- Nuxt by default only prerenders static routes!

```
Built at: 05/14/2019 8:54:46 PM
Asset Size Chunks Chunk Names
69222603641ec14958f1.js 3.48 KiB 4 [emitted] pages/posts/_id
7a99eda2137e1ca28867.js 1.24 KiB 3 [emitted] pages/index
8c0eba4f9d3c357ace19.js 957 bytes 5 [emitted] pages/seo
b9e4dd90c12d65d1e2e8.js 3.46 KiB 1 [emitted] pages/PostList
f5e068d9fad922baff60.js 1.12 KiB 2 [emitted] pages/history
server.js 389 KiB 0 [emitted] app
server.manifest.json 723 bytes [emitted]
+ 6 hidden assets
Entrypoint app = server.js server.js.map
i Generating pages 20:54:46
✓ Generated /PostList 20:54:48
✓ Generated /history 20:54:48
✓ Generated /seo 20:54:48
✓ Generated / 20:54:48
MacBook-Pro:800-nuxt PeterKassenaar$
```

# Rendering dynamic routes



- Add configuration to nuxt.config.js
- Static, like this:

Dynamic routes are ignored by the `generate` command.

Example:

```
-| pages/
---| index.vue
---| users/
-----| _id.vue
```

Only the route `/` will be generated by Nuxt.js.

If you want Nuxt.js to generate routes with dynamic params, you need to set an array of dynamic routes.

We add routes for `/users/:id` in `nuxt.config.js`:

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





# ...or dynamically

```
import posts from './data/posts';
```

```
package.json × _id.vue × PostList.vue × default.vue × nuxt.config.js × seo.vue × index.vue × posts.js ×
65 },
66 /**
67 * Generate options, https://nuxtjs.org/api/configuration-generate/ (PK)
68 * TODO: create dynamic function, returning a promise
69 */
70 generate: {
71 routes: function (callback) {
72 const routes = posts.map(post => {
73 return `/posts/${post.id}`
74 });
75 callback(null, routes)
76 }
77 }
78 }
```

# Static pages generated for GoogleBot

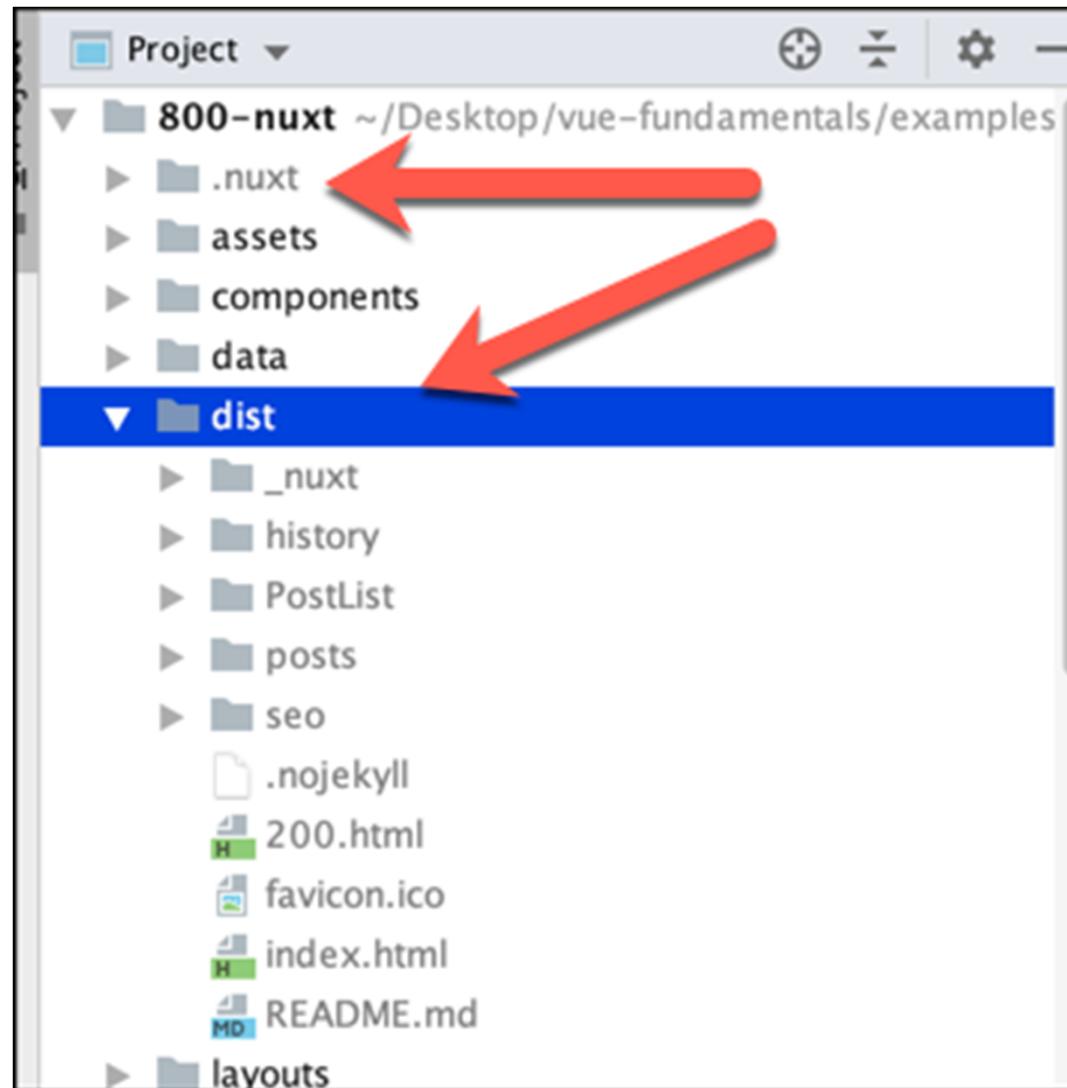


```
Terminal: Terminal × Terminal (2) × +
+ 6 hidden assets
Entry point app = server.js server.js.map
i Generating pages
✓ Generated /history
✓ Generated /PostList
✓ Generated /seo
✓ Generated /
✓ Generated /posts/2
✓ Generated /posts/5
✓ Generated /posts/3
✓ Generated /posts/4
✓ Generated /posts/7
✓ Generated /posts/6
✓ Generated /posts/9
✓ Generated /posts/12
✓ Generated /posts/8
✓ Generated /posts/10
```

A red arrow points from the text "Generated /posts/2" in the terminal output towards the bottom left of the slide.

<https://nuxtjs.org/api/configuration-generate>

# Deploy the /dist directory (.nuxt not!)



# Deploying to a server

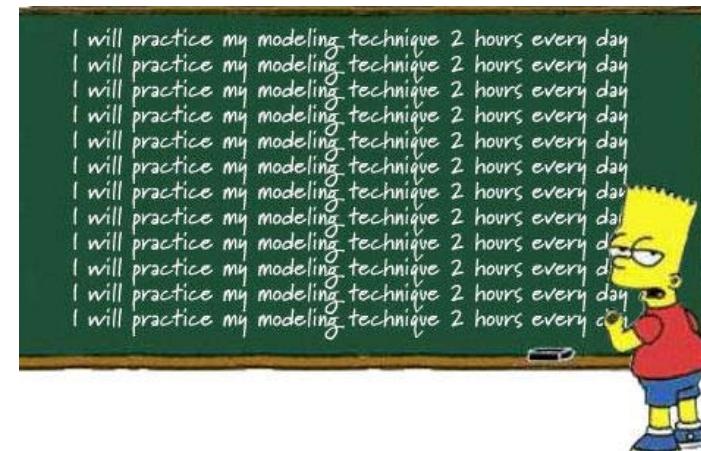


- Deploy to Heroku
- Deploy to Netlify
- Deploy to IIS
- <https://nuxtjs.org/guide/commands>

# Workshop



- Generate a production build from your application
  - Use `npm run generate`
- Read the deployment documentation at  
<https://nuxtjs.org/guide/commands>
- Complete example `../800-nuxt`



# Checkpoint



- ...