

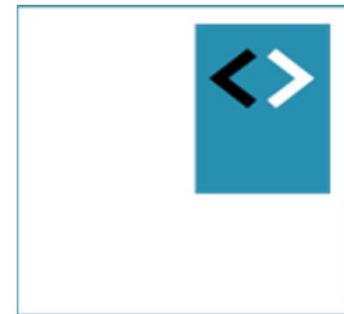


Vue Fundamentals

01 - Introduction



nationale
nederlanden



Peter Kassenaar –
info@kassenaar.com

Peter Kassenaar

- Trainer, author, developer – sinds 1996
- Specialty: “*Everything JavaScript*”
- JavaScript, ES6, Angular, NodeJS, TypeScript, jQuery, PhoneGap, Vue, etc.

www.kassenaar.com/blog

info@kassenaar.com

Twitter: [@PeterKassenaar](https://twitter.com/@PeterKassenaar)





World-class Vue training in Dutch and English

Live classrooms - focused on today's developers

LEARN MORE

www.vuetraining.nl/

github.com/PeterKassenaar/nn

The screenshot shows the GitHub repository page for `PeterKassenaar / nn`. The repository has 1 star, 0 forks, and 0 issues. It contains 1 commit, 1 branch, and 0 releases. There is 1 contributor. The latest commit is `c5bc682` from just now. The README.md file contains the text "Slides and extra links on the training VueJS at Nationale Nederlanden Rotterdam, april 2019".

Pull requests Issues Marketplace Explore

PeterKassenaar / nn

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Slides and extra links on the training VueJS at Nationale Nederlanden Rotterdam, april 2019 Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

PeterKassenaar Initial commit Latest commit `c5bc682` just now

README.md Initial commit just now

README.md

nn

Slides and extra links on the training VueJS at Nationale Nederlanden Rotterdam, april 2019

About you...



Tell us a little bit about yourself

Knowledge of Vue, other front-end frameworks?

Have you worked with HTML, CSS, JavaScript,

TypeScript and/or other (web)languages?

Tell us a little bit about your projects.

What are your expectations of this course?

Material

Software (Vue + libraries)

Handouts (PDF, Github)

Workshops (Github)

Websites (online)



vuejs.org/

Agenda - 3 days

- Introduction – overview of the front-end landscape
- Vue tooling – installation
- Hello World –the structure and architecture of Vue apps.
- Zooming in:
 - Components, data binding, attribute binding, event binding

Agenda – cont'd

- Styling components
- Using Mixins
- Communication between components
 - Props and events
- Vue router
- Route parameters

Agenda – cont'd

- Managing state with **Vuex**
- Communicating with **external API's** and live data
- **Deploy** your app to production

Labs and example code

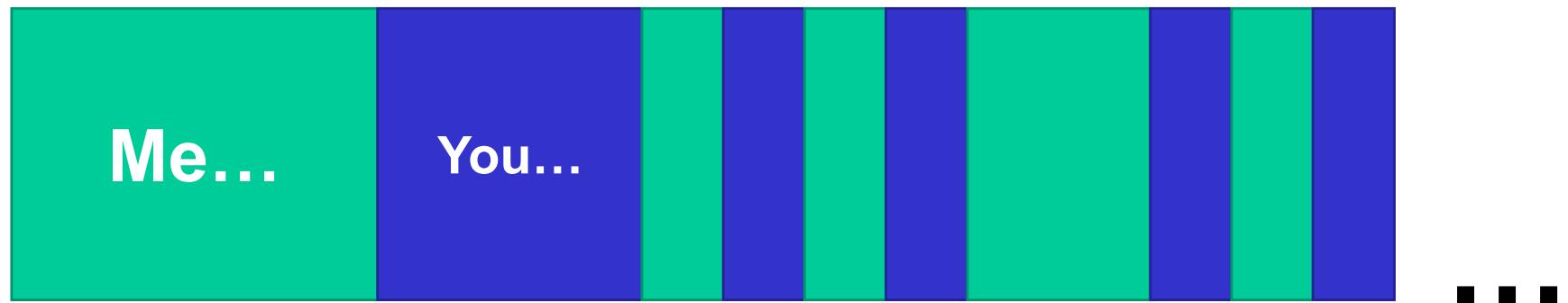
1. Labs/Exercises

- In the PDF's in the Github-repo. But: feel free to deviate. Adapt to suit your own needs! (hobby, work, current projects)

2. Example code

- Executions of the exercises, small projects (`npm install`, `npm run serve`)
- Work in progress – let me know of additions/errors!
- github.com/PeterKassenaar/vue-fundamentals

Overall process



Questions?

Addressing the “WHY” question!

WHY, would we want to use a frontend framework.

It is all HTML, CSS and JavaScript right?

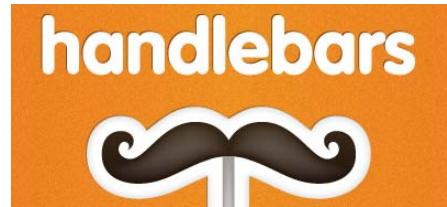
Rethorical question:

“Do we want to go back

to the jQuery days?”

Old school web apps

HTML + templates



Data Binding



Routing



DOM-manipulation



Mobile development

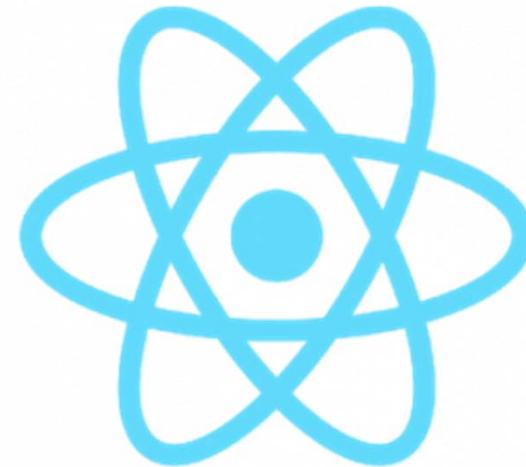


...

“The Frankenstein Framework”



Front-end Frameworks – the big four



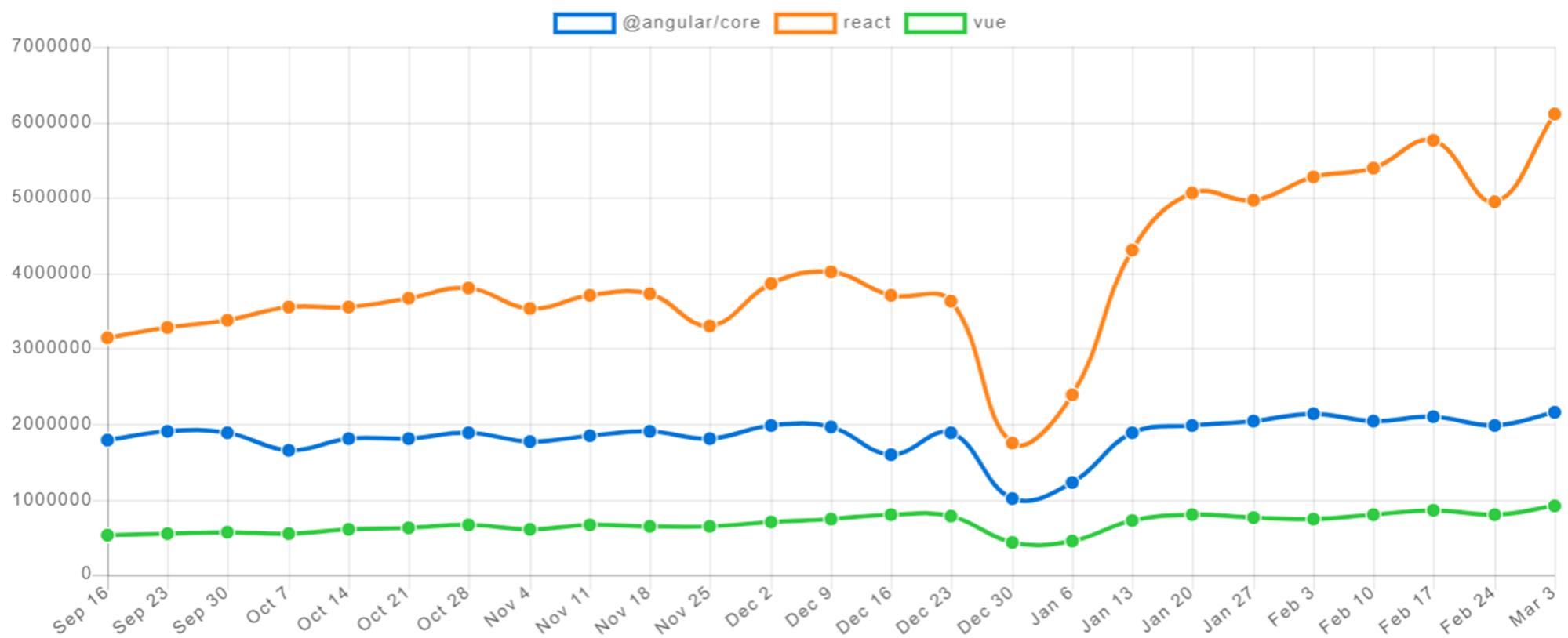
Vue.js – Created by Evan You

The screenshot shows Evan You's Patreon page for the Vue.js project. At the top, there's a navigation bar with 'PATREON' on the left, a search bar with 'Find a creator', and links for 'Create On Patreon', 'Explore Creators', 'Sign Up', and 'Log In'. Below the header is a large circular badge for Vue.js, featuring a green gradient background with a white 'V' in the center. The text 'FINEST QUALITY JAVASCRIPT' is curved along the top edge, and 'EST. 2014' is at the bottom. To the left of the badge is a circular profile picture of Evan You wearing Google Glass. Below the badge, the text 'Evan is creating Vue.js' is displayed. The main content area has tabs for 'Overview' (which is active) and 'Posts'. A prominent red button on the right says 'BECOME A PATRON'. On the left, a sidebar shows '227 patrons' and '\$16,646 per month'. It includes social sharing buttons for Facebook and Twitter, and links for 'SHARE' and 'TWEET'. Below this is a 'GOALS' section with a 'View all' link and navigation arrows. The central content area features a large green and blue 'V' logo, followed by two paragraphs of text about invoicing and one-time donations. At the bottom, it says 'Vue.js is a front-end framework that is simple to use yet powerful in nature.' and 'The project focuses on making some of the best ideas in web UI development (components, declarative UI, hot-reloading, time-travel...'. On the right, there's a 'TIERS' section with two levels: 'Backer' (\$10 or more per month) and 'Generous Backer' (\$50 or more per month). Each tier has a description and a 'JOIN \$10 TIER' or 'JOIN \$50 TIER' button.

No backing from company with deep pockets – all crowd funded

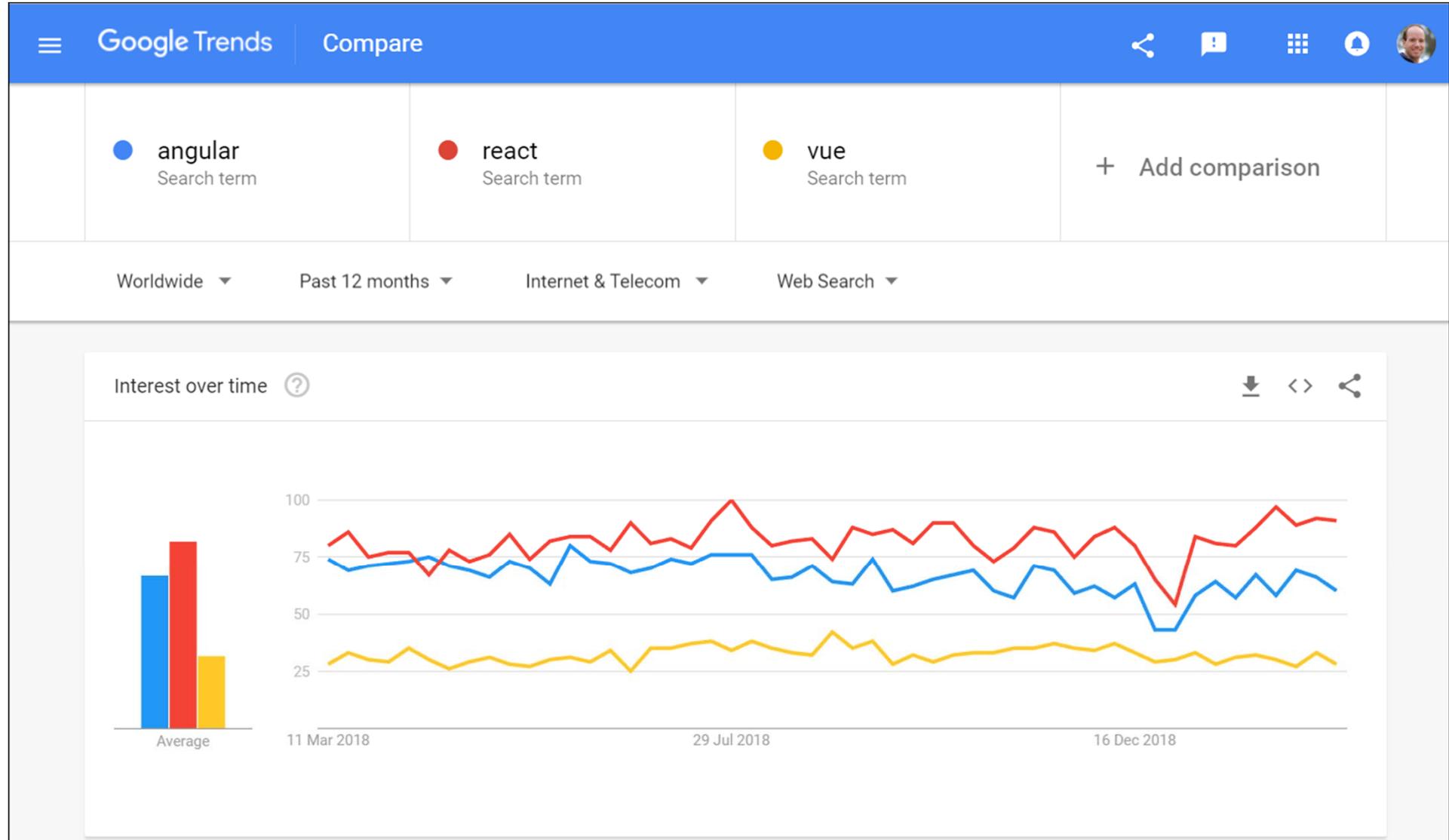
A 2019 comparison

Downloads in past 6 Months ▾



<https://www.npmtrends.com/@angular/core-vs-react-vs-vue>

Google Trends



<https://trends.google.com/trends/explore?cat=13&q=angular,react,vue>

Giststar rankings

Gitstar Ranking Users Organizations Repositories GitHub username Search Sign in with GitHub

Repositories Ranking

1 2 3 ... >

(Δ) 1. freeCodeCamp/freeCodeCamp	★ 298396
▼ 2. vuejs/vue	★ 129161
(B) 3. twbs/bootstrap	★ 128833
(F) 4. facebook/react	★ 124937
↑TF 5. tensorflow/tensorflow	★ 123450
Ebook 6. EbookFoundation/free-programming-books	★ 118765
OWL 7. sindresorhus/awesome	★ 99459
D3 8. d3/d3	★ 83265
zeit/next.js	★ 35904
moment/moment	★ 35598
nodejs/node-v0.x-archive	★ 35577
ionic-team/ionic	★ 35555
typicode/json-server	★ 35389
mrdoob/three.js	★ 35013
rust-lang/rust	★ 34749
shadowsocks/shadowsocks-windows	★ 34747

<https://gitstar-ranking.com/repositories>



unicorn.supplies

[Follow](#)Jens Neuhaus [Follow](#)

CTO to rent, Full-stack developer with Python/Django, supporting startups & mid-size companies, loves go...

Aug 28 · 25 min read

Angular vs. React vs. Vue: A 2017 comparison

Deciding on a JavaScript framework for your web application can be overwhelming. [Angular](#) and [React](#) are very popular these days, and there is an upstart which has been getting a lot of traction lately: [VueJS](#). What's more, these are just a few of the [new kids on the block](#).



4.5K

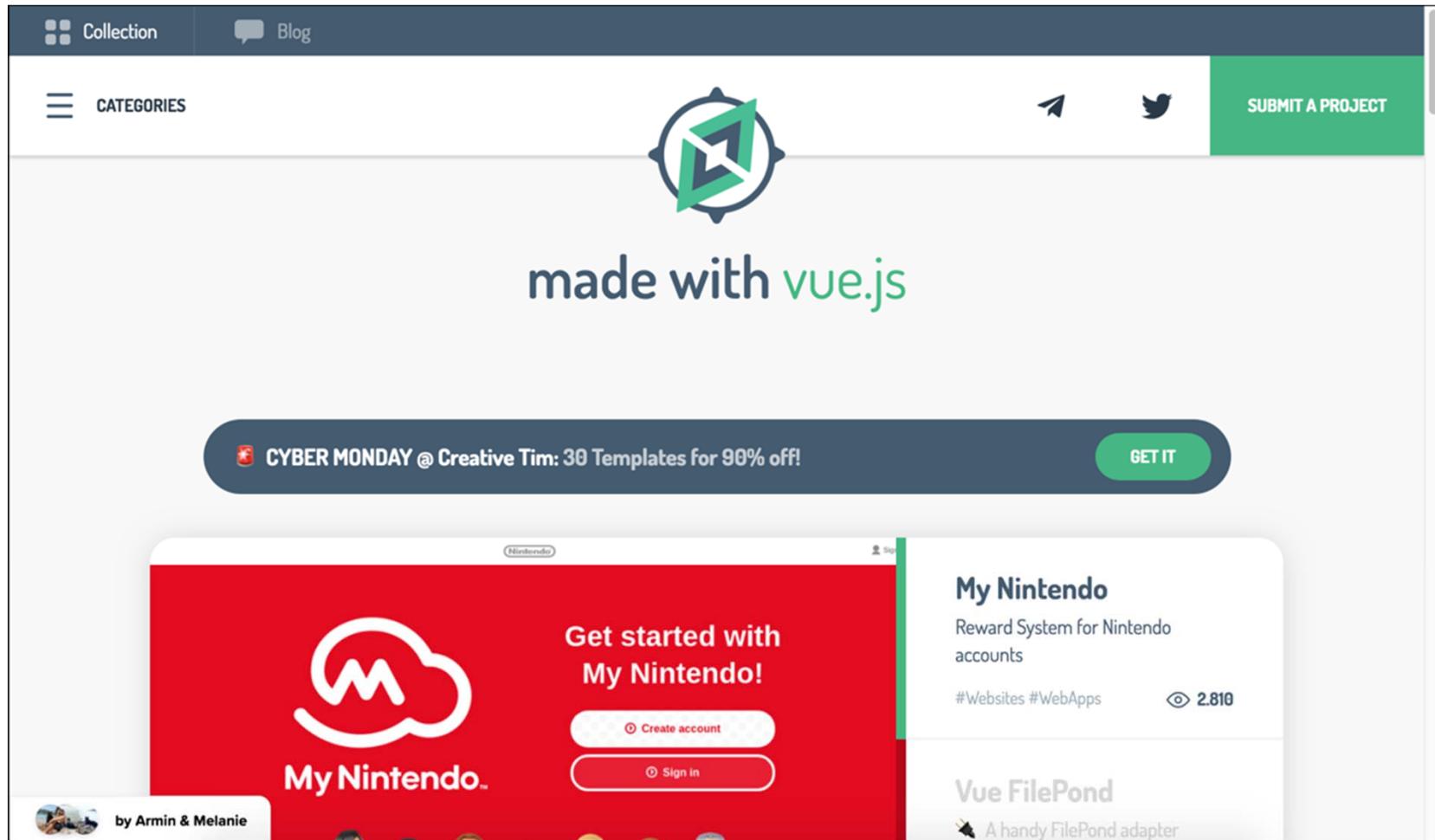


38

Next story
Moving From Angular to Vue : A ...

<https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176>

“Vue is used by Alibaba, Baidu, Nintendo, GitLab, L’Oréal and many others”



<https://madewithvuejs.com/>



MijnOverheid

Activeer uw MijnOverheid account

Activeer uw account met uw DigiD. Hulp nodig? Bekijk de instructievideo

Activeren



MijnOverheid

Berichtenbox

Lopende zaken

Persoonlijke gegevens

Inloggen

Ervaar het gemak van MijnOverheid

Via MijnOverheid heeft u toegang tot uw post en persoonlijke gegevens van steeds meer overheidsorganisaties. U kunt via MijnOverheid bijvoorbeeld controleren hoe u bij de gemeente geregistreerd staat of de status van uw omgevingsvergunning volgen. Ook ontvangt u hier post van organisaties zoals de Belastingdienst, Kadaster, RDW, SVB, UWV en uw gemeente. U logt veilig en gemakkelijk in met uw DigiD.

Do you have a European approved digital identity? [Please login here.](#)

De Berichtenbox app van MijnOverheid

Post van de overheid direct lezen op uw mobiel of tablet? Dat kan met de Berichtenbox app van

Bekijk de video

Veelgestelde vragen

Ik wil weten of mijn MijnOverheid-account is
geactiveerd

Wat is MijnOverheid?

Hoe herken ik valse e-mails?

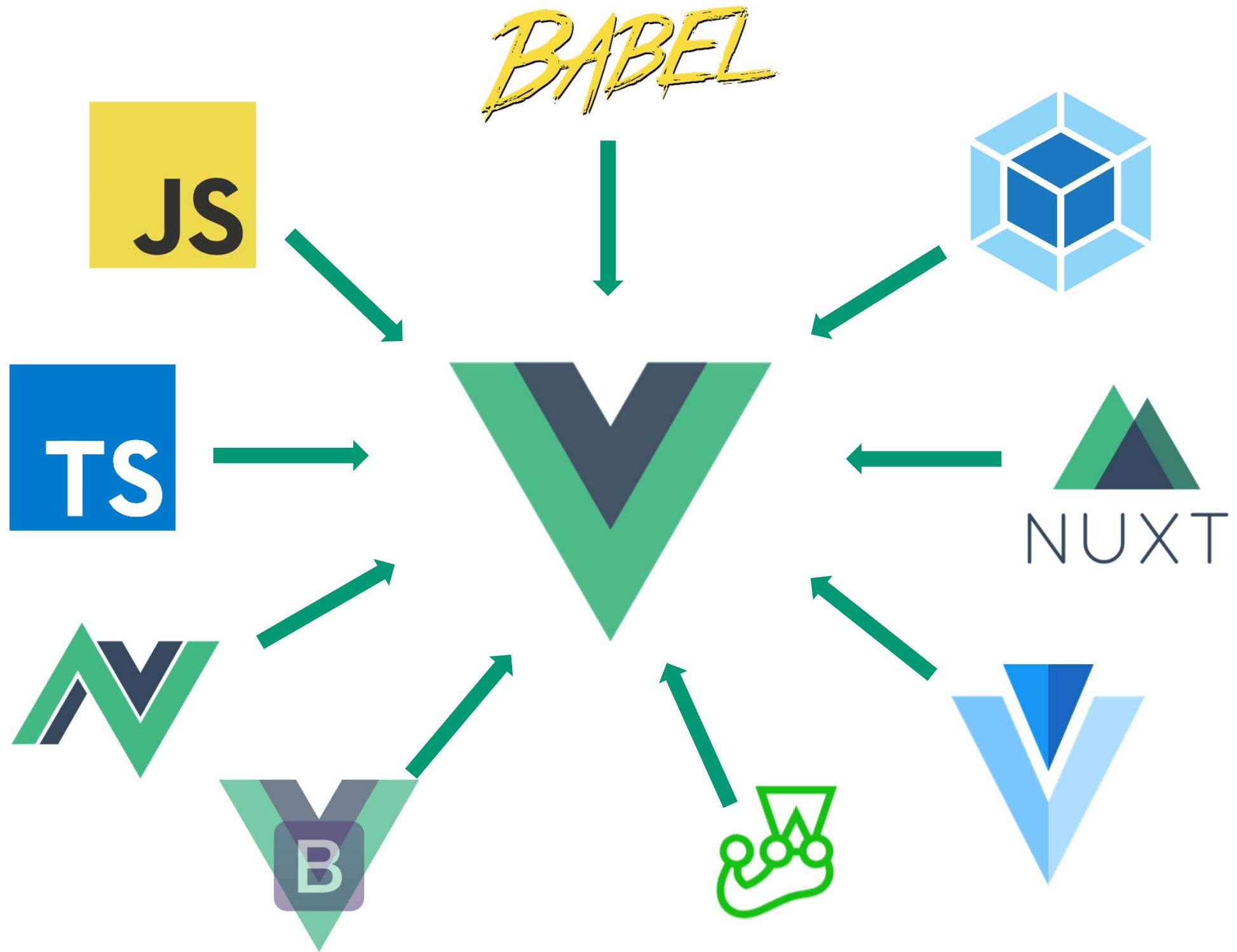
Hoe voorkom ik phishing?

Wat is er nodig om gebruik te kunnen maken van
MijnOverheid?



The Vue.js ecosystem

Vue is not isolated/is not living on an island



Additional services, options, etc.

The image displays four separate web application interfaces side-by-side:

- Amplify Framework**: An orange-themed landing page featuring the text "The foundation for your cloud-powered mobile & web apps". It includes a "GET STARTED" button and icons for Apple, Android, and the web.
- Cloudinary**: A dark-themed landing page with the Cloudinary logo and navigation links for "About", "Blog", "Contact", "SOLUTIONS", "PRICING", and "CLOUDINARY".
- Vue Mastery**: A light blue-themed landing page for learning Vue.js, with the text "The ultimate guide for Vue". It has "Courses" and "Pricing" links.
- CodeSandbox**: A dark-themed code editor landing page with the text "The online code editor for Web". It features a large "OPEN EDITOR" button and a "EXPLORE EXAMPLES" link.

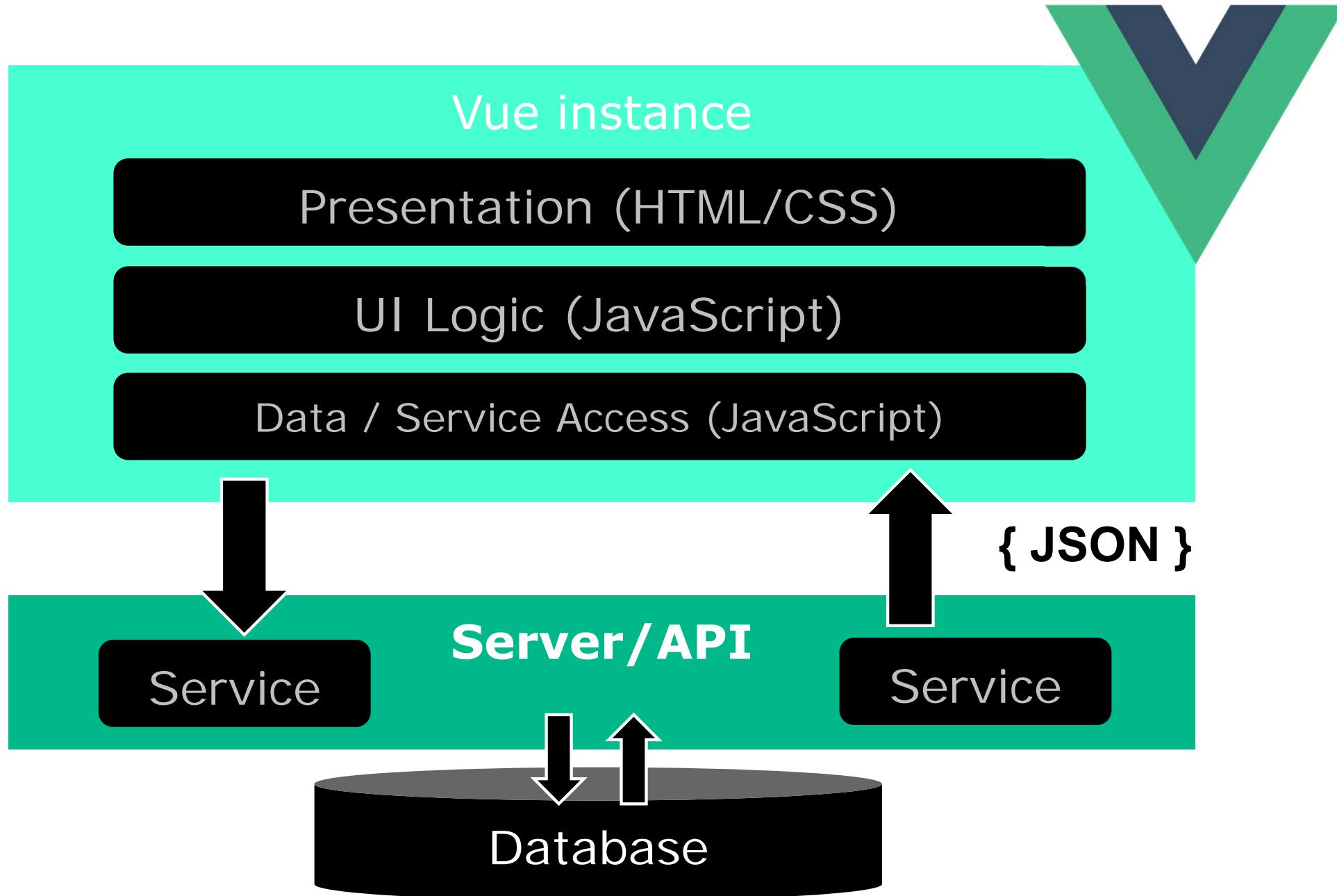
And more....



Vue application structure

Getting a conceptual understanding of the Vue app architecture

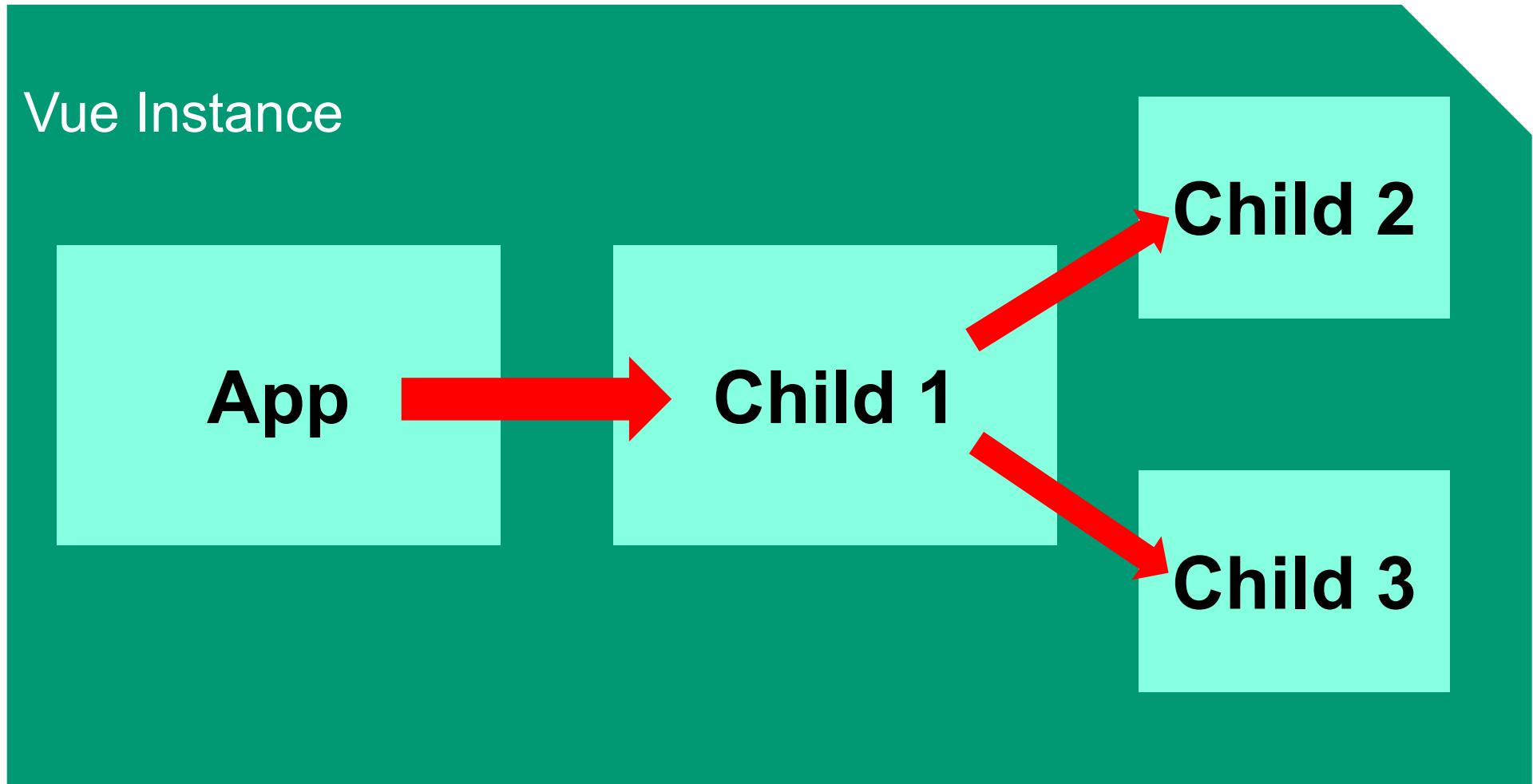
Single Page Application



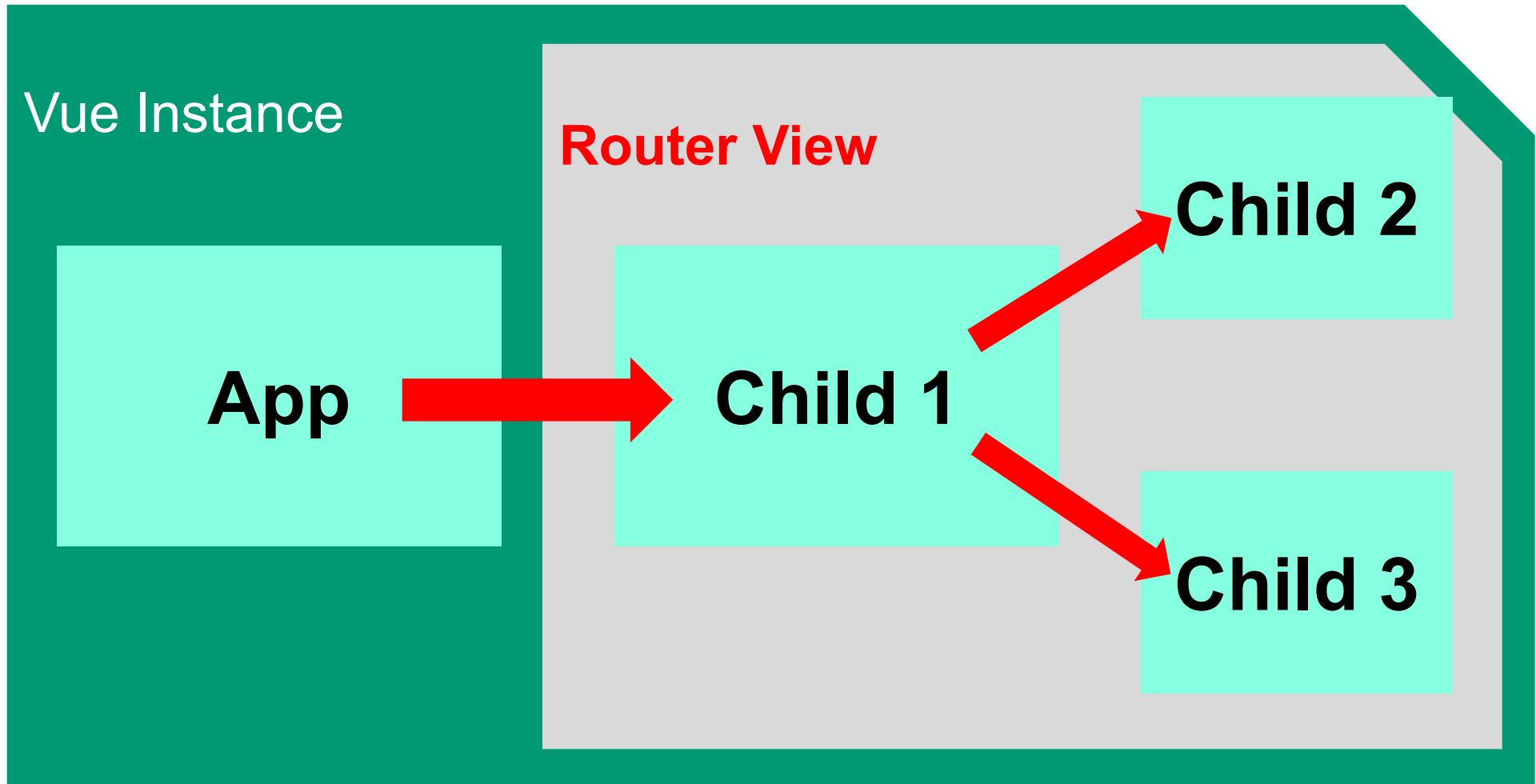
The Vue instance

- The heart of every application : the **Vue instance**
- Created whenever you call `new Vue()` in your code
- Typical application: just 1 (one) Vue instance
- Pass in a configuration object
- Render a component. Typically by calling
`new Vue({render : h => h(App)})`

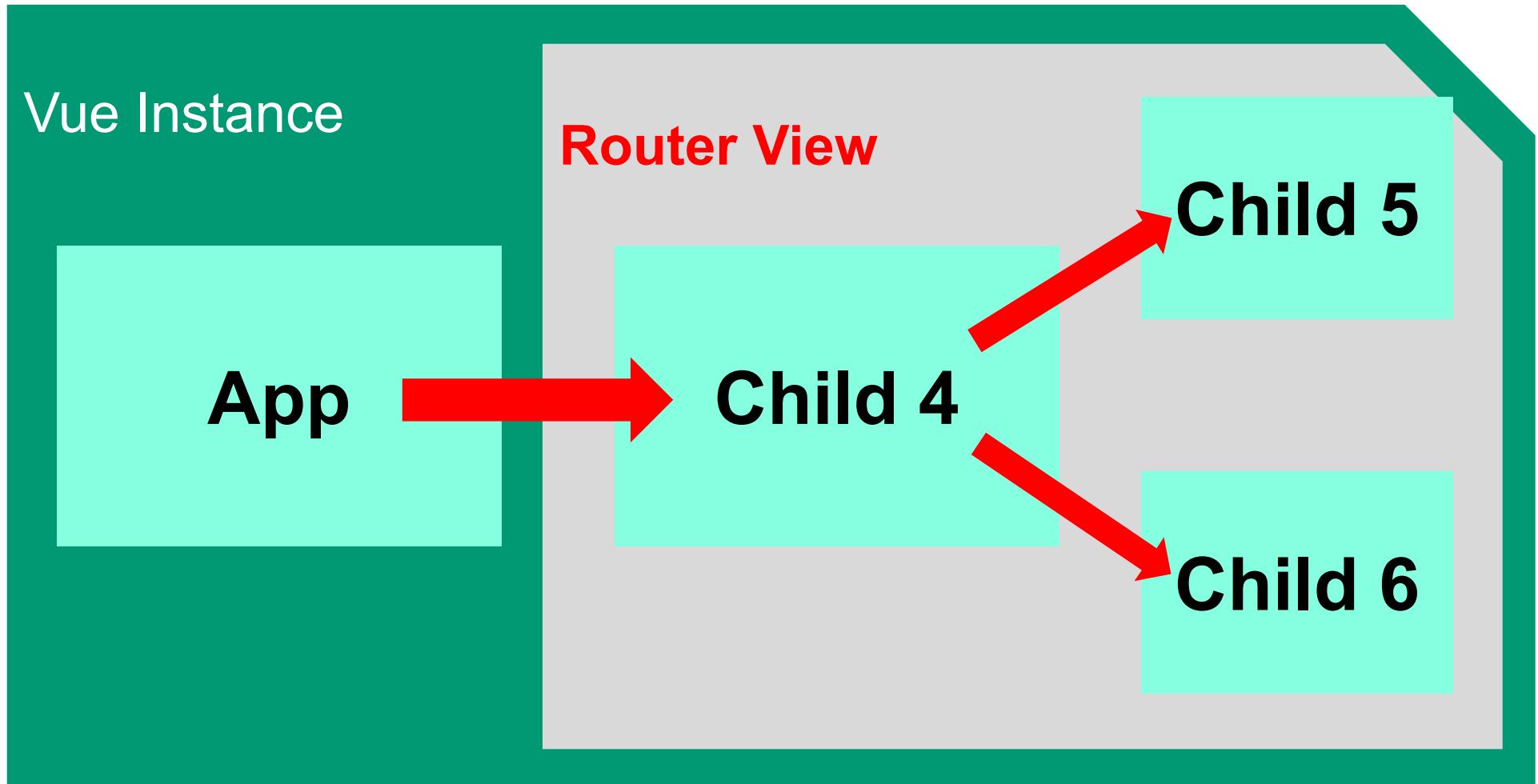
Visual of the Vue instance



Vue app with routing



As the URL changes, content gets updated





Let's create an app

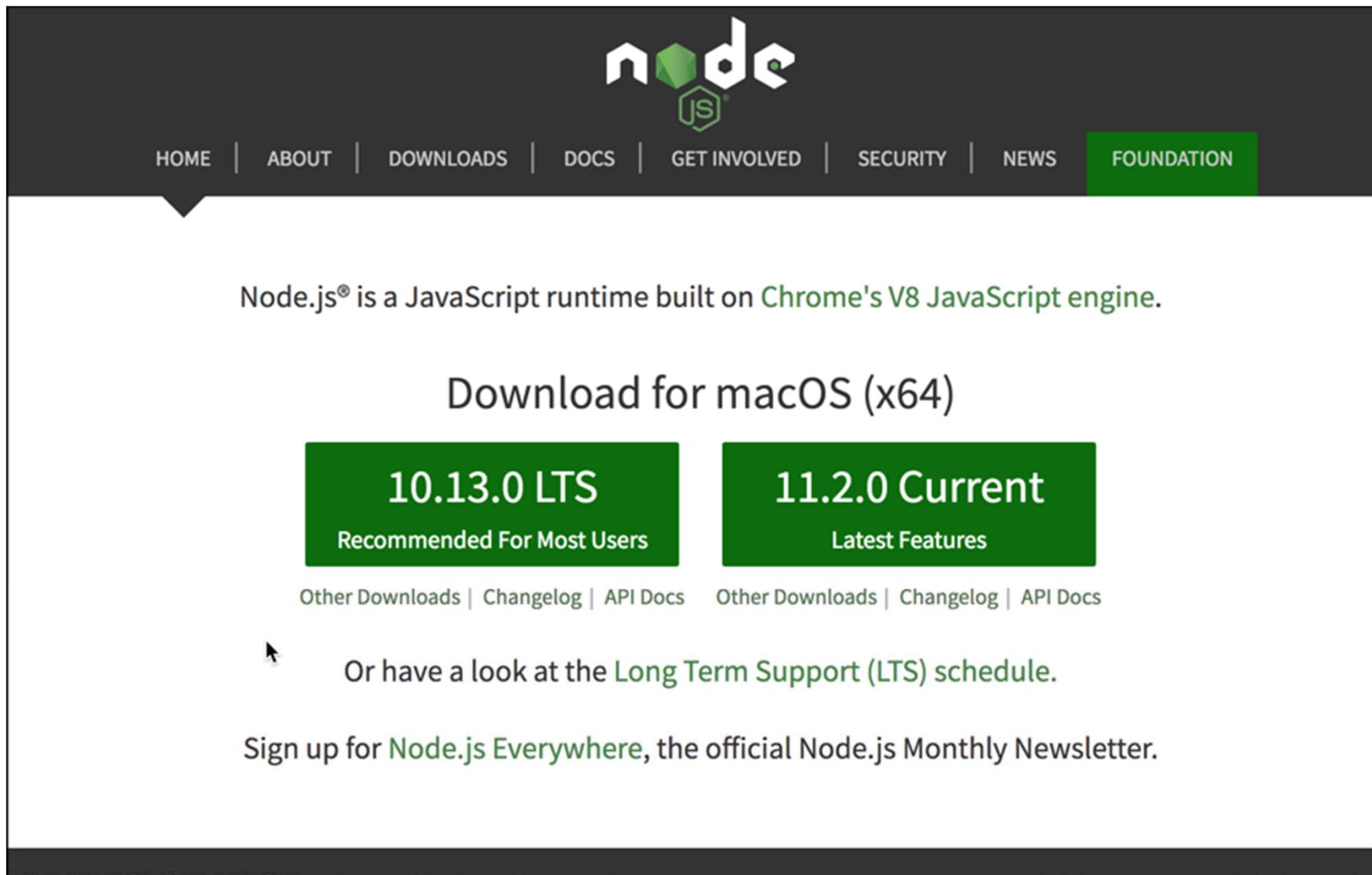
Install tooling and create your first app

Install necessary Tooling

First – install Git, if you don't have it yet

The screenshot shows the official Git website at <https://git-scm.com/downloads>. The main navigation bar includes links for About, Documentation, Downloads (selected), GUI Clients, Logos, and Community. A sidebar mentions the "Pro Git book" by Scott Chacon and Ben Straub. The central "Downloads" section features sections for Mac OS X, Windows, and Linux/Unix. A prominent callout highlights the "Latest source Release 2.19.2" with a "Download 2.19.0 for Mac" button. Below this, a monitor icon displays a teal screen with the same release information. In the bottom right corner of the monitor, there's a user profile for "PeterKassenaar". To the right of the monitor, a terminal window is open with the command `git --version` run, showing the output "git version 2.17.2 (Apple Git-113)". The output "git version 2.17.2" is circled in red.

NodeJS



The screenshot shows the official Node.js website. At the top is a dark navigation bar with the Node.js logo and links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. The FOUNDATION link is highlighted with a green background. Below the navigation is a large white area containing text and download buttons. The text reads: "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." Below this is a heading "Download for macOS (x64)". Two green rectangular buttons are displayed side-by-side: the left one is labeled "10.13.0 LTS" and "Recommended For Most Users"; the right one is labeled "11.2.0 Current" and "Latest Features". Below each button are links to "Other Downloads", "Changelog", and "API Docs". A small mouse cursor icon points towards the LTS button. Further down, there is a link to the "Long Term Support (LTS) schedule" and a call to action to "Sign up for Node.js Everywhere".

<https://nodejs.org/en/>

Vue CLI

- Scaffold new projects
- Add routing
- Add State management and so on
- Answer prompts when creating a new project
- <https://cli.vuejs.org/>

```
npm install -g @vue/cli
```

cli.vuejs.org

Vue CLI 3 

Guide Config Reference Plugin Dev Guide ▾ Plugins ▾ Changelog Languages ▾ GitHub ▾



Vue CLI 3

🛠 Standard Tooling for Vue.js Development

Get Started →

Always check the documentation

The screenshot shows the Vue CLI 3 documentation website. The top navigation bar includes links for Guide (which is underlined in green), Config Reference, Plugin Dev Guide, Plugins, Changelog, Languages, and GitHub. The left sidebar has a 'Overview' section selected, followed by Installation, Basics (Instant Prototyping, Creating a Project, Plugins and Presets, CLI Service), Development (Browser Compatibility, HTML and Static Assets, Working with CSS, Working with Webpack, Environment Variables and Modes), and Build Targets. The main content area features a heading 'Overview'. Below it is a message: 'This project is sponsored by  Share Code'. A yellow warning box contains the text: 'WARNING This documentation is for `@vue/cli` version 3.x. For the old `vue-cli`, see [here](#)'. The main text explains that Vue CLI is a full system for rapid Vue.js development, providing various features listed in a bulleted list.

This project is sponsored by  Share Code

WARNING

This documentation is for `@vue/cli` version 3.x. For the old `vue-cli`, see [here](#).

Vue CLI is a full system for rapid Vue.js development, providing:

- Interactive project scaffolding via `@vue/cli`.
- Zero config rapid prototyping via `@vue/cli` + `@vue/cli-service-global`.
- A runtime dependency (`@vue/cli-service`) that is:
 - Upgradeable;
 - Built on top of webpack, with sensible defaults;
 - Configurable via in-project config file;
 - Extensible via plugins
- A rich collection of official plugins integrating the best tools in the frontend ecosystem.
- A full graphical user interface to create and manage Vue.js projects.

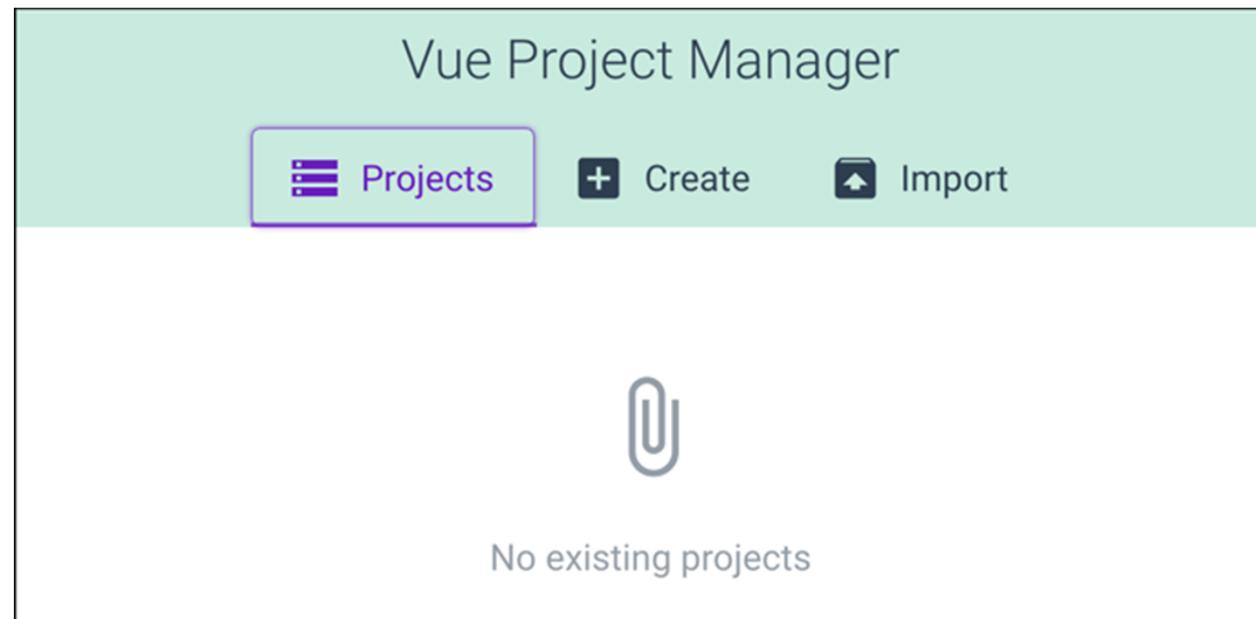
<https://cli.vuejs.org/guide/>

Installation of @vue/cli

```
[MacBook-Pro:~ PeterKassenaar$ npm install -g @vue/cli
npm WARN deprecated hoek@5.0.4: This version is no longer maintained. Please upgrade to the latest version.
npm WARN deprecated cross-spawn-async@2.2.5: cross-spawn no longer requires a build toolchain, use it instead
[MacBook-Pro:~ PeterKassenaar$ vue --version
3.1.3
MacBook-Pro:~ PeterKassenaar$ ]
```

Creating a project

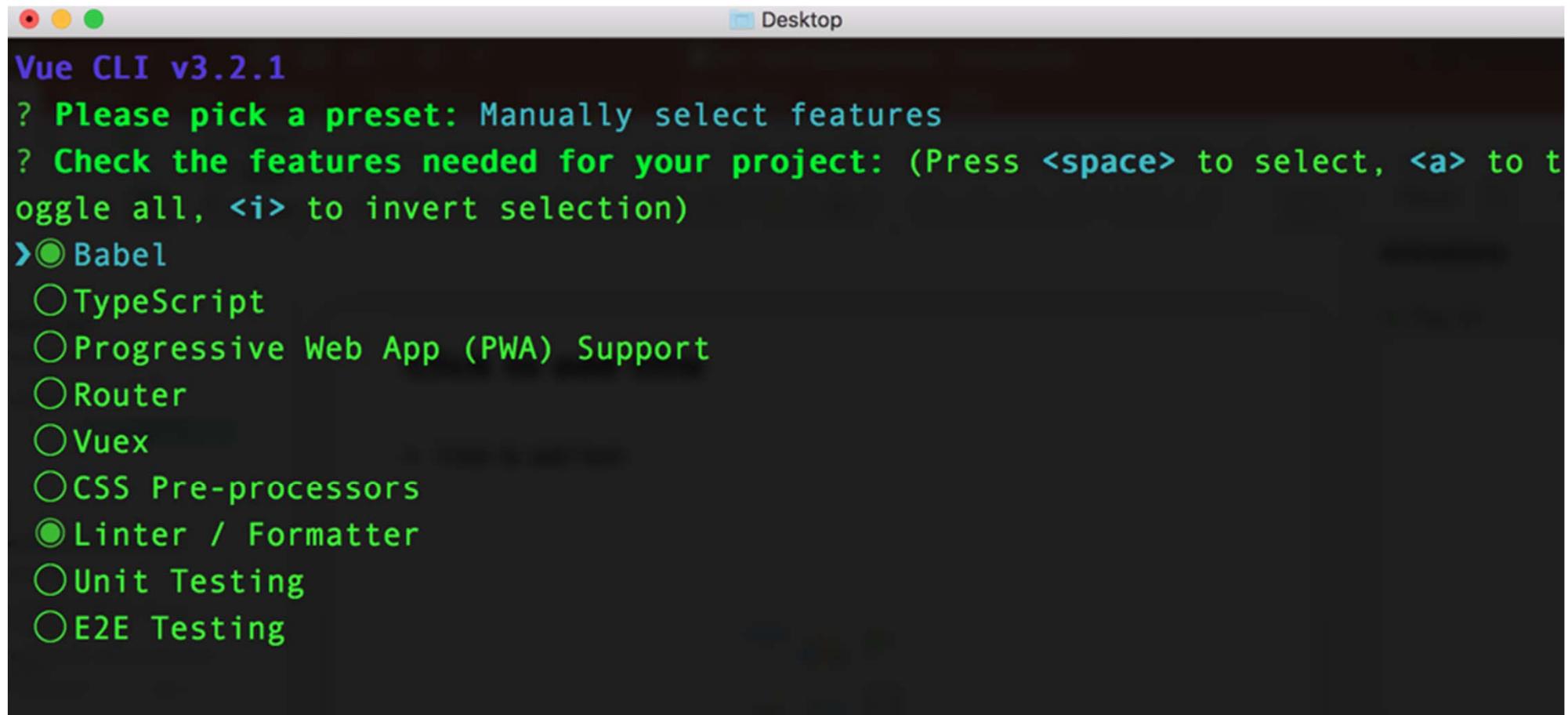
- Using the command line
 - `vue create my-vue-app`
- Using the GUI
 - `vue ui`



We're using the command line here

- Main command
 - `Vue create <app-name>`
- The CLI creates a folder with name for you
- Answer the prompt/questions
 - `Manually select features`
- Plugins can create / add their own features and questions
- You can add stuff (`router`, `vuex`) later on.

```
vue create my-vue-project
```



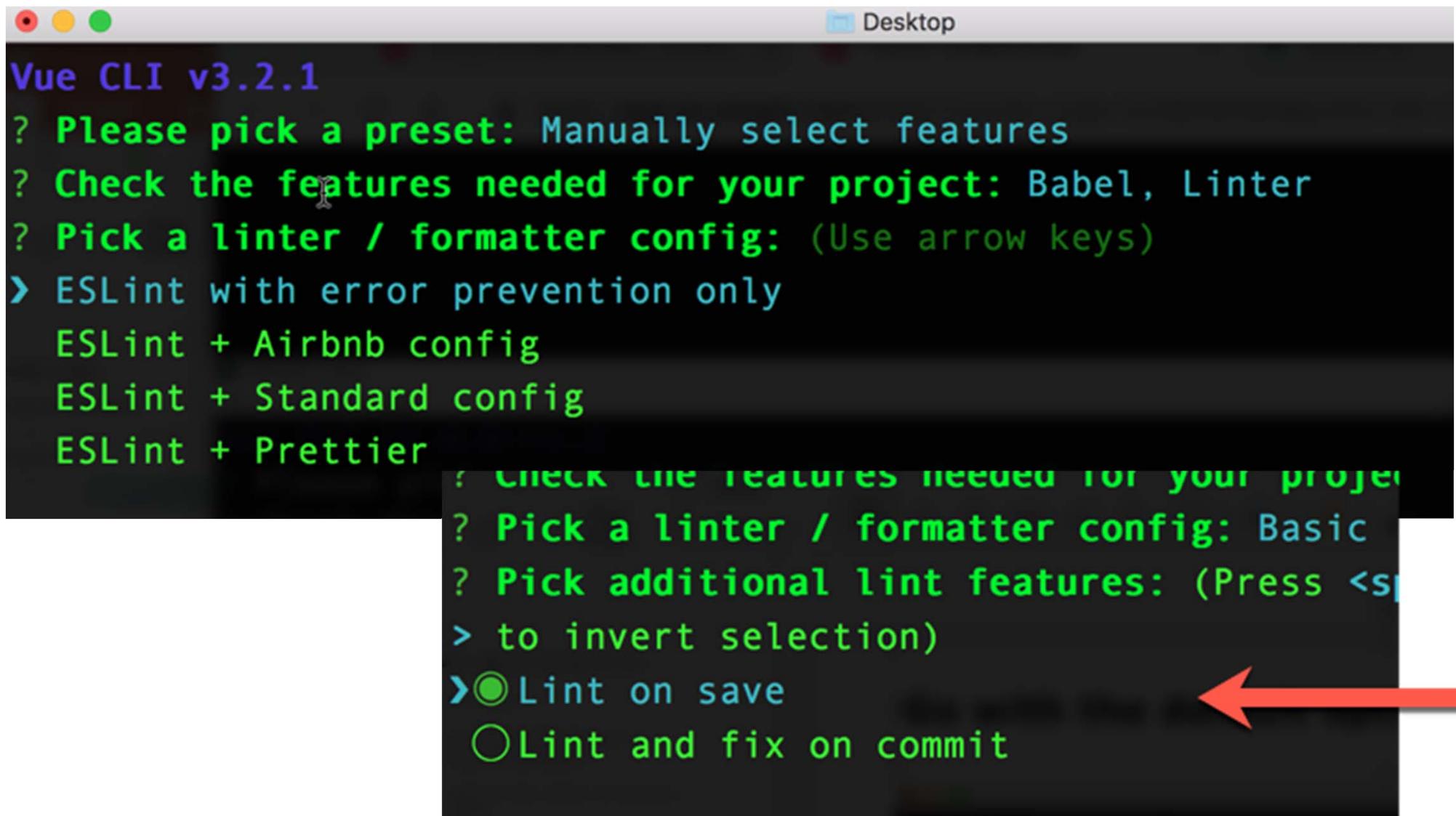
The screenshot shows a terminal window titled "Desktop" running on a Mac OS X system. The title bar has red, yellow, and green buttons. The terminal displays the following text:

```
Vue CLI v3.2.1
? Please pick a preset: Manually select features
? Check the features needed for your project: (Press <space> to select, <a> to t
oggle all, <i> to invert selection)
❯◉ Babel
○ TypeScript
○ Progressive Web App (PWA) Support
○ Router
○ Vuex
○ CSS Pre-processors
◉ Linter / Formatter
○ Unit Testing
○ E2E Testing
```

The "Babel" option is selected (indicated by a filled circle). The "Router" and "Vuex" options are also listed as available features.

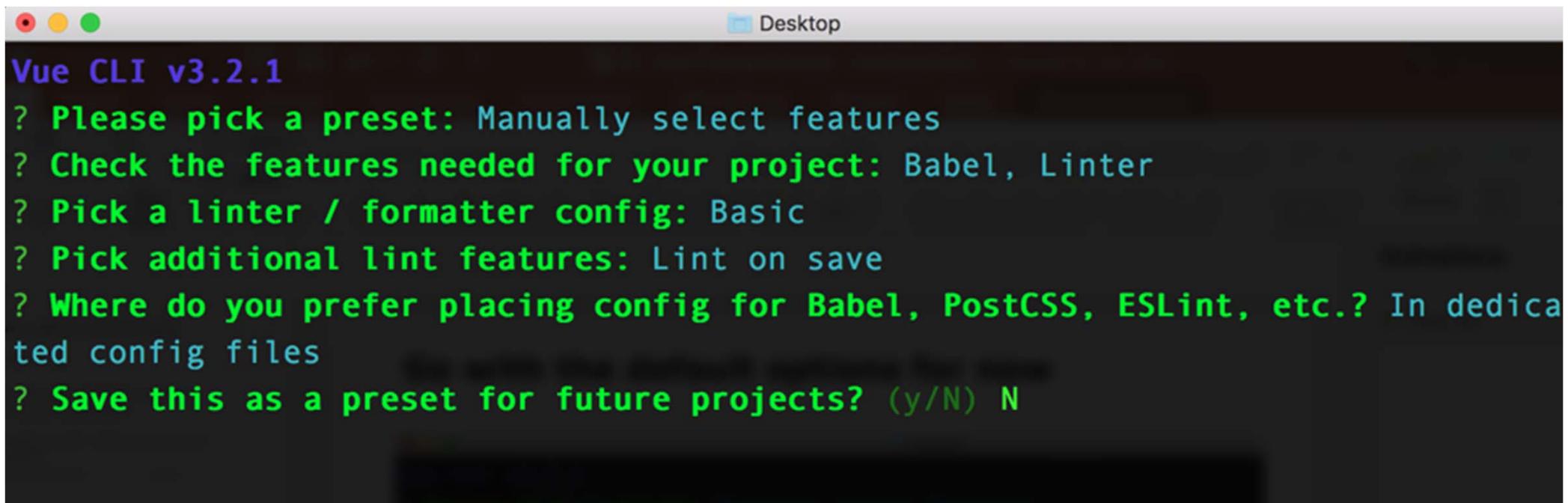
We're going to add Router and Vuex later

Go with the default options for now



```
Vue CLI v3.2.1
? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, Linter
? Pick a linter / formatter config: (Use arrow keys)
❯ ESLint with error prevention only
  ESLint + Airbnb config
  ESLint + Standard config
  ESLint + Prettier
    ? Check the features needed for your project
      ? Pick a linter / formatter config: Basic
      ? Pick additional lint features: (Press <space> to invert selection)
        >● Lint on save
        ○ Lint and fix on commit
```

Our options



Vue CLI v3.2.1

- ? Please pick a preset: Manually select features
- ? Check the features needed for your project: Babel, Linter
- ? Pick a linter / formatter config: Basic
- ? Pick additional lint features: Lint on save
- ? Where do you prefer placing config for Babel, PostCSS, ESLint, etc.? In dedicated config files
- ? Save this as a preset for future projects? (y/N) N

Creating the project...

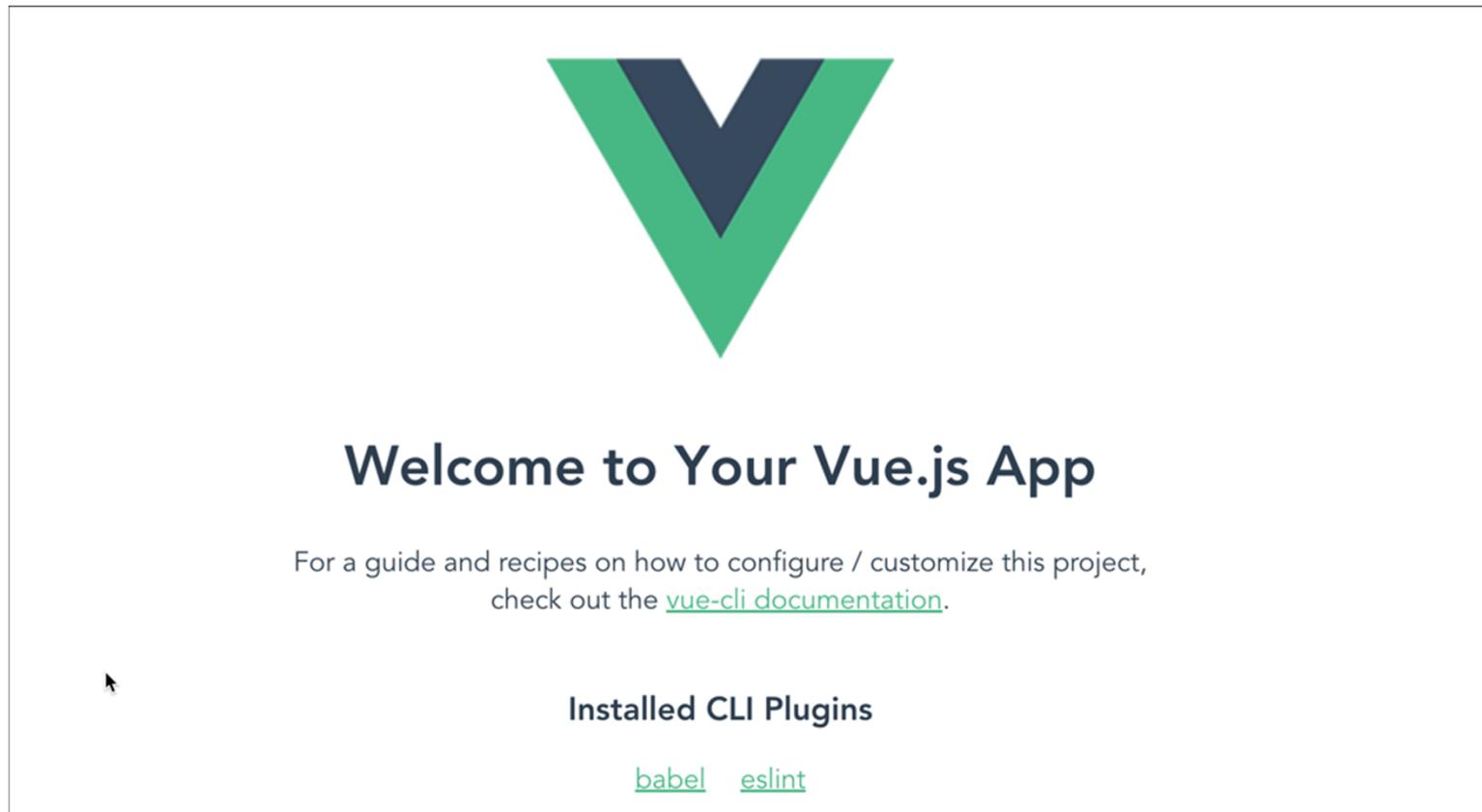
The screenshot shows a terminal window on a Mac OS X desktop. The window title is "Desktop". The terminal content is as follows:

```
Vue CLI v3.2.1
  Creating project in /Users/PeterKassenaar/Desktop/my-vue-app.
  Initializing git repository...
  Installing CLI plugins. This might take a while...

yarn install v1.10.1
info No lockfile found.
[1/4] 🔎 Resolving packages...
· @babel/helper-plugin-utils@^7.0.0
```

Standaardproject starten

- npm run serve
- <http://localhost:8080>



Saved presets?

`~/.vuerc`

“Saved presets will be stored in a JSON file named `.vuerc` in your user home directory. If you wish to modify saved presets / options, you can do so by editing this file.”

Configuration options

```
vue create --help
```

sh

```
Usage: create [options] <app-name>
```

```
create a new project powered by vue-cli-service
```

Options:

-p, --preset <presetName>	Skip prompts and use saved or remote preset
-d, --default	Skip prompts and use default preset
-i, --inlinePreset <json>	Skip prompts and use inline JSON string as preset
-m, --packageManager <command>	Use specified npm client when installing dependencies
-r, --registry <url>	Use specified npm registry when installing dependencies
-g, --git [message false]	Force / skip git initialization, optionally specify message
-n, --no-git	Skip git initialization
-f, --force	Overwrite target directory if it exists
-c, --clone	Use git clone when fetching remote preset
-x, --proxy	Use specified proxy when creating project
-b, --bare	Scaffold project without beginner instructions
-h, --help	Output usage information

<https://cli.vuejs.org/guide/creating-a-project.html#vue-create>



Let's look at the code

A look inside the key files in the generated code

Package.json

```
{  
  "name": "my-vue-app",  
  "version": "0.1.0",  
  "private": true,  
  "scripts": {  
    "serve": "vue-cli-service serve",  
    "build": "vue-cli-service build",  
    "lint": "vue-cli-service lint"  
  },  
  "dependencies": {  
    "vue": "^2.5.17"  
  },  
  "devDependencies": {  
    "@vue/cli-plugin-babel": "^3.2.0",  
    "@vue/cli-plugin-eslint": "^3.2.0",  
    "@vue/cli-service": "^3.2.0",  
    "babel-eslint": "^10.0.1",  
    "eslint": "^5.8.0",  
    "eslint-plugin-vue": "^5.0.0-0",  
    "vue-template-compiler": "^2.5.17"  
  }  
}
```

Main.js

```
import Vue from 'vue'  
import App from './App.vue'  
  
Vue.config.productionTip = false;  
  
new Vue({  
  render: h => h(App),  
}).$mount('#app');
```

(productionTip shows a message in the console if the app runs in production code; you can safely switch this off)

App.vue

- The AppComponent is the main component of the app.
 - It generally doesn't change
- Look at the component
- Three sections:
 - Template
 - Script
 - Style
- For now it only loads the logo and the HelloWorld component

App.vue

```
<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App" name="Peter"/>
  </div>
</template>
```

Template

```
<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>
```

Script

```
<style>
#app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  ...
}
</style>
```

Style

HelloWorld Component

- HelloWorld is the actual ‘homepage’ of the generated project.
- Its name is HelloWorld – look that up!
- It has a *property* called msg
 - ...which is filled from the calling component App.vue. Have a look at that!
 - It holds most of the html, links, etc.
- Have a look at the <styles scoped> section.
 - We'll cover scoped styles later
- People generally delete this component and replace it with their own main component

On separation of concerns...

“Separation of concerns is not equal to separation of file types.”

...

“you can still leverage its hot-reloading and pre-compilation features by separating your JavaScript and CSS into separate files”

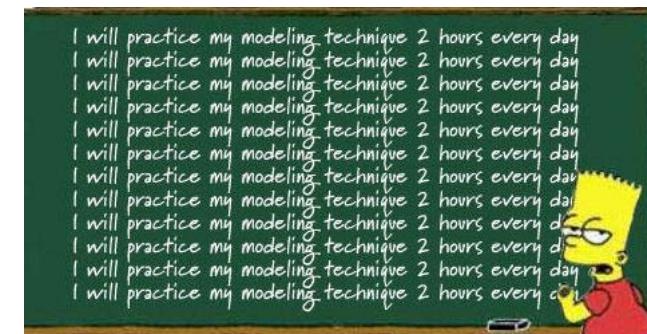
HTML

```
<!-- my-component.vue -->
<template>
  <div>This will be pre-compiled</div>
</template>
<script src=".my-component.js"></script>
<style src=".my-component.css"></style>
```

<https://vuejs.org/v2/guide/single-file-components.html>

Workshop

- Install @vue/cli
 - <https://cli.vuejs.org/>
- Create a new, blank Vue project
- Identify and study the files discussed in this section
 - Run the project: `npm run serve`
- Replace the HTML in HelloWorld component with your own HTML
- Add a new property to HelloWorld and give it some content
- Code example: [..../100-helloworld](#)





Components

Adding and using your own components

Types of components

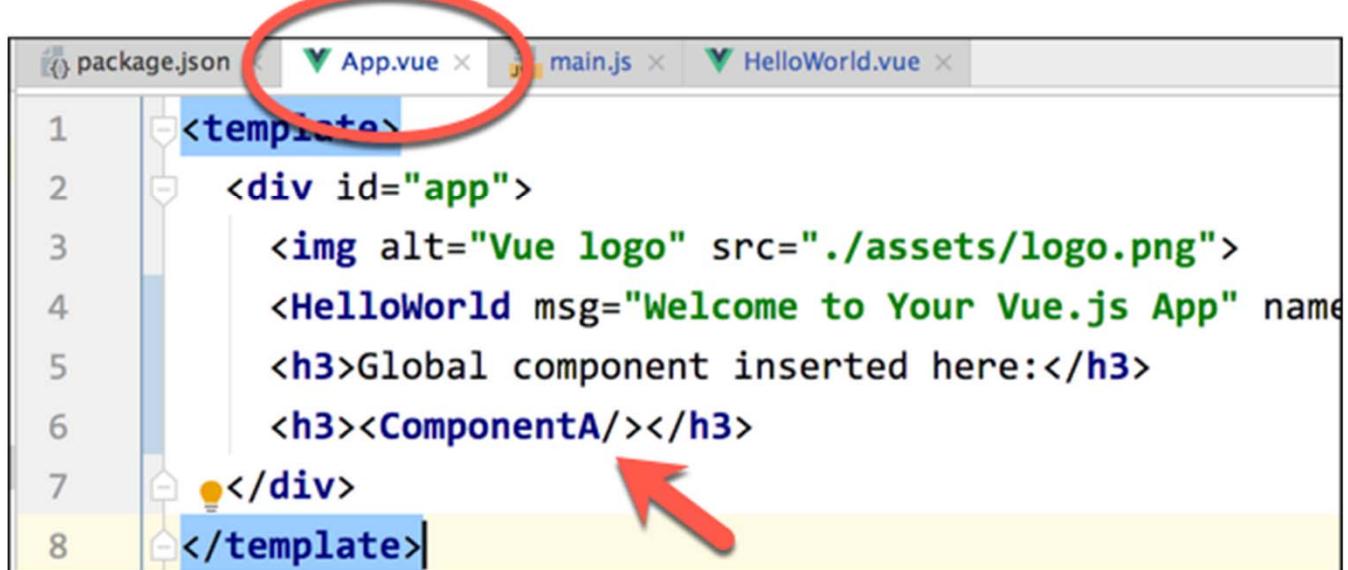
Global Components

**Single File
Components**

Global components



```
package.json x App.vue x main.js x HelloWorld.vue x
1 import Vue from 'vue'
2 import App from './App.vue'
3
4 // creating a global component
5 // (you don't do this very often in real applications)
6 Vue.component('ComponentA', {
7   template: `<span>Hello World - this is component A</span>`
8 });
9
```



```
package.json x App.vue x main.js x HelloWorld.vue x
<template>
1 <div id="app">
2   
3   <HelloWorld msg="Welcome to Your Vue.js App" name="Vidya">
4     <h3>Global component inserted here:</h3>
5     <h3><ComponentA/></h3>
6   </div>
7 </template>
8
```

Downside of global components

- They register *global variables* – chances of naming conflicts
- Inline HTML, all string templates – no color coding etc in some editors
- CSS is not encapsulated
- No build time compiling support for templates

Verdict – try to avoid global components

Single File components

The screenshot shows a code editor with several tabs at the top: package.json, App.vue, SingleFileComponent.vue (which is the active tab), main.js, and HelloWorld.vue. The SingleFileComponent.vue tab has a red circle around it. The code in the editor is:

```
1 <template>
2   <div>Template should have a single element as its root</div>
3 </template>
4
5 <script>
6   // single file components should have a name
7   export default {
8     name: "SingleFileComponent"
9   }
10 </script>
11
12 <style scoped>
13   /*Generally, styles are scoped*/
14 </style>
15
```

Three specific sections of the code are highlighted with red circles:

- The entire `<template>` block.
- The entire `<script>` block.
- The entire `<style scoped>` block.

A yellow bar is visible at the bottom of the code editor window.

Single File Components

- Files that have a .vue extension
- Three sections
 - <template> - required
 - <script> - required
 - <style> - optional
- Some prefer the <script> section first.
 - Use a style guide in your organization!
- SFC's are precompiled by WebPack at buildtime

Verdict – use Single File Components

Using a SFC

The screenshot shows a code editor with several tabs at the top: package.json, App.vue, SingleFileComponent.vue, main.js, and HelloWorld.vue. The main content is the App.vue file, which contains the following code:

```
5      <h3>Single File Component inserted here:</h3>
6      <SingleFileComponent/> ←
7    </div>
8  </template>
9
10 <script>
11   import HelloWorld from './components/HelloWorld.vue'
12   import SingleFileComponent from "./components/SingleFileComponent"; ←
13
14   export default {
15     name: 'app',
16     components: {
17       SingleFileComponent, ←
18       HelloWorld
19     }
20   }

```

Three red arrows point to the following parts of the code:

- An arrow points to the opening tag of the SingleFileComponent in the template.
- An arrow points to the import statement for SingleFileComponent.
- An arrow points to the component entry in the components object.

Import a SFC and use it's name in the template

Best practices on components

- Use Two words
 - to avoid naming conflicts with standard HTML elements, which are always one word.
- Use UpperCamelCase on naming components
- Group components inside their own folder, preferably inside /components
- <https://vuejs.org/v2/style-guide/>



Vue.js



Learn

Ecosystem

Team

Support Vue

Priority A: Strongly Recommended

Priority C: Recommended

Priority D: Use with Caution

Priority A Rules: Essential

Multi-word component names

Component data

Prop definitions

Keyed v-for

Avoid v-if with v-for

Component style scoping

Private property names

Priority B Rules: Strongly Recommended

Component files

Single-file component filename casing

Base component names

Single-instance component names

Tightly coupled component names

Order of words in component names

Self-closing components

Priority A Rules: Essential (Error Prevention)

Multi-word component names ESSENTIAL

Component names should always be multi-word, except for root **App** components.

This **prevents conflicts** with existing and future HTML elements, since all HTML elements are a single word.

Bad

```
Vue.component('todo', {  
  // ...  
})
```

JS

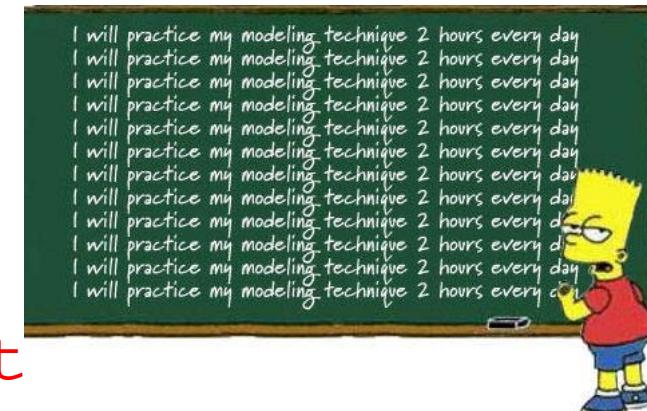
```
export default {  
  name: 'Todo',  
  // ...  
}
```

JS

<https://vuejs.org/v2/style-guide/#Component-files-strongly-recommended>

Workshop

- Add two components to your application
- Use Single File Components
 - Place them in the /components folder
- Give them some UI
- See if you can get scoped styles to work
 - (i.e. create a style with the same name, but different properties in the components)
- Include them in the HelloWorld component
- Example [.../102-multiple-component](#)



Checkpoint

- You know what the Vue Instance is
- You know how to install Vue CLI and what its purpose is
- You know how to generate new Vue projects
- You can identify the key files and explain / edit them
- You know about components
- You can create Single File Components (SFC's) and include them in the app