



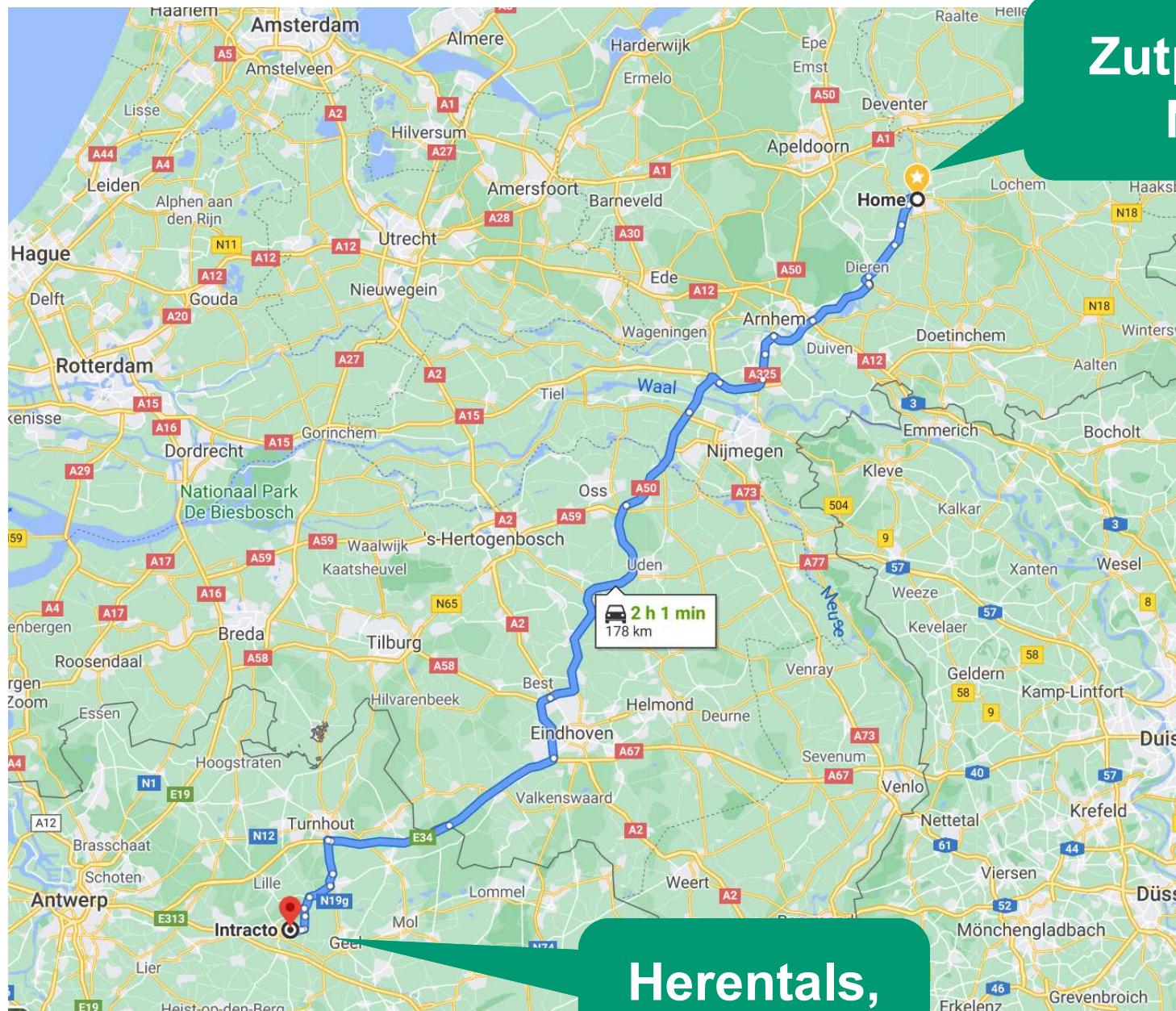
Vue Fundamentals

01 - Introduction



Peter Kassenaar –
info@kassenaar.com

Zutphen,
NL



Herentals,
BE

Peter Kassenaar

- Trainer, author, developer – since 1996
- Specialty: "*Everything JavaScript*"
- JavaScript, ES6, Angular, NodeJS, TypeScript, jQuery, React, Vue, etc.

www.kassenaar.com

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/intracto

The screenshot shows the GitHub repository page for `PeterKassenaar/intracto`. The repository has 1 branch and 0 tags. It contains four files: `.gitignore`, `LICENSE`, `README.md`, and a commit from `PeterKassenaar` titled "Initial commit". The `Code` button is highlighted. The repository has 1 commit, 1 pull request, and 0 issues. The `Code` section shows the contents of `README.md`, which includes the text "Slides and example code on the training Vue.js + NodeJS - Intracto, sept. 2020". The repository has 0 stars and 0 forks.

PeterKassenaar / intracto

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Octotree

master 1 branch 0 tags

Go to file Add file Code

About

Slides and example code on the training Vue.js + NodeJS - Intracto, sept. 2020

Releases

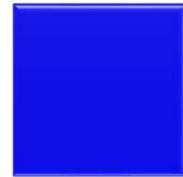
No releases published Create a new release

Packages

No packages published Publish your first package

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

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 (Node.js, Vue.js + libraries, editor)
- Handouts (PDF, Github)
- Workshops (Inside Presentations, Github)
- Websites (online)



vuejs.org/

Global Agenda



14-15-16 Sept. 2020 – Mo. + Tue. + Wed.

~09:00 start

~ 12:00-13:00 Lunch Break

~16:15-16:30 wrap-up

Wednesday – probably wrap-up a little bit early

Agenda - 3 days



Day 1 – Vue.js Fundamentals

Day 2 – Vue.js Advanced Concepts

Day 3 – Node.js

Agenda - day 1

- Vue Fundamentals
- Overview of the Vue **frontend landscape** (concise)
- **Architecture and structure** of Vue applications
- Data binding with `{{ ... }}`
- Attribute binding with `v-bind:` or `:`
- Event binding with `v-on:` or `@`
- Vue directives (`v-model`, `v-if`, `v-show`, etc)
- Component communication with **props** and **events**
- ...



Day 2 – Vue - Advanced concepts



- Routing
- Routing parameters
- State management with Vuex
- Composition API
- Introduction Nuxt & SSR
- ...

Day 3 – Node.js

- Make a choice...
- Node architecture and concepts
- Relationship with NPM
- Npm init and more npm CLI commands
- Structure and role of package.json
- Creating your own Node.js applications/packages
- Sample Node Core modules (Timers, File System, Http, Path, etc)
- Express + Express Router
- MongoDB
- Deployment
- Publication on NPM or Nexus ...

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

github.com/PeterKassenaar/vue-fundamentals

The screenshot shows the GitHub repository page for 'PeterKassenaar / vue-fundamentals'. The repository has 73 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. The latest commit was on April 15. The README.md file contains the following content:

```
vue-fundamentals

example code on the training Vue Fundamentals by Peter Kassenaar

Manage topics

- 73 commits
- 1 branch
- 0 packages
- 0 releases
- 1 contributor

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

PeterKassenaar Added TypeScript example on Composition API using RestCountries API | Latest commit 59303bd on Apr 15
examples Added TypeScript example on Composition API using RestCountries API | 2 months ago
.gitignore updated .gitignore | 2 years ago
JavaScript APIs.txt updated example 410 vuex | 16 months ago
README.md Updated README.md | 3 months ago

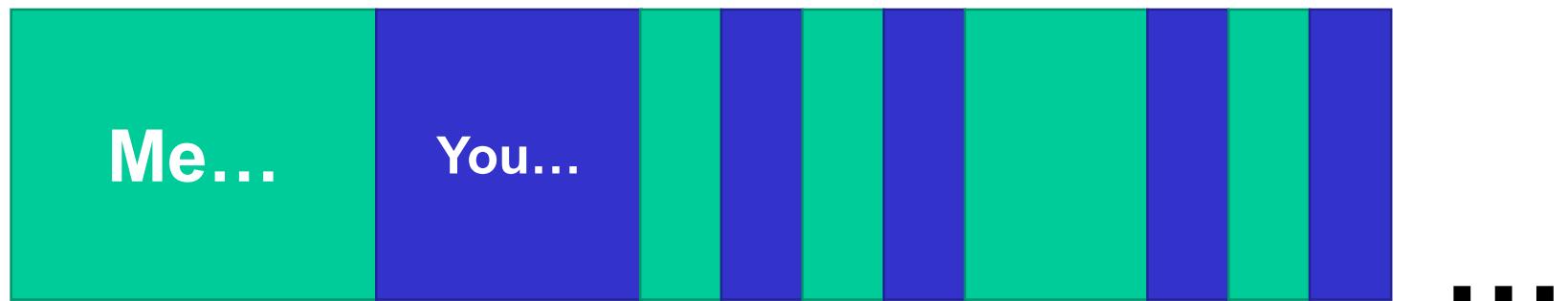
README.md
```

vue-fundamentals

example code on the training Vue Fundamentals.

Usage

Overall process



Questions?

Addressing the “WHY” question!

WHY, would we want to use a frontend framework.

It is all **HTML, CSS** and **JavaScript** right?

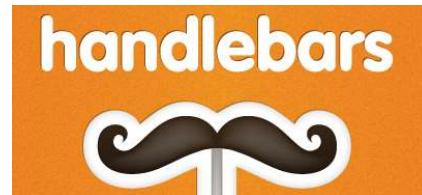
Rethorical question:

**“Do we want to go back
to the jQuery days?”**

speed, not re-inventing the wheel,
consistency,,
community,
performance,
testing....

Old school web apps

HTML + templates



Data Binding



Routing



DOM-manipulation



Mobile development

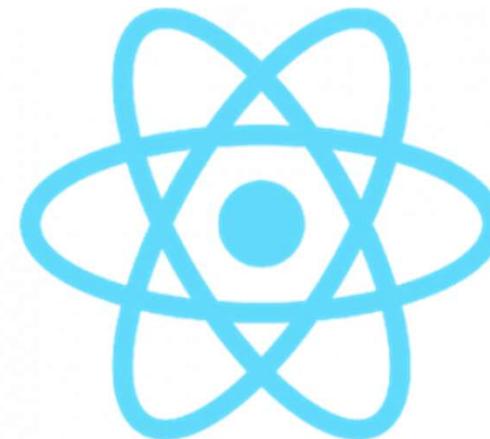


...

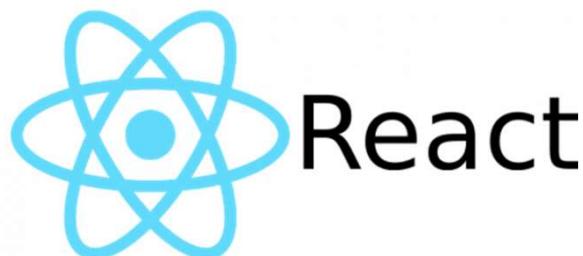
“The Frankenstein Framework”



Front-end Frameworks – the big four



Similarities

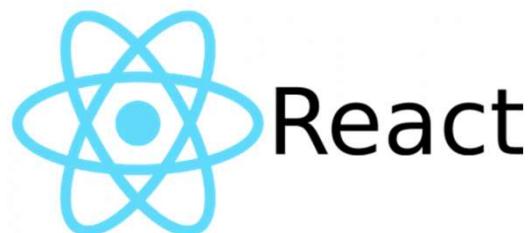


- Creating Single Page Applications
- Based on components
- Data binding, props, events, routing, state management, ...
- Huge ecosystem
- Huge community
- High adaptation rate

Differences (apart from syntax)



- Point of departure: **HTML template**, enhanced with framework specific tags and attributes
- One-stop-shop / solution



- Point of departure: **JavaScript**, **JavaScript**, **JavaScript (JSX)**
- Build-all-yourself

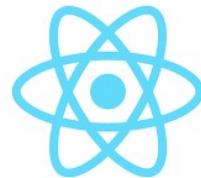
Frameworks vs. Libraries



- Framework (Angular, Vue)
- Concise (less code)
- Template centered
- Separate files
- Non-standard (html, elements, shadow DOM)
- Community backing (except Angular)



- React: Library
- Explicit (more coding)
- JavaScript Centered
- Single File
- Non standard (JSX, shadow DOM)
- Corporate backing



Vue.js – Created by Evan You

PATREON | Find a creator | Create On Patreon | Explore Creators | Sign Up | Log In

Evan is creating Vue.js

OVERVIEW

227 patrons

\$16,646 per month

+ FOLLOW SHARE

TWEET

SHARE

GOALS View all < >

If you need invoices for tax purposes, simply message me on Patreon with your company information after pledging and I can setup automatic invoicing (for Gold Sponsors and up only).

If you are not comfortable with recurring pledges, I also accept one-time donations via PayPal or crypto currencies :)

Vue.js is a front-end framework that is simple to use yet powerful in nature.

The project focuses on making some of the best ideas in web UI development (components, declarative UI, hot-reloading, time-travel

BECOME A PATRON

TIERS

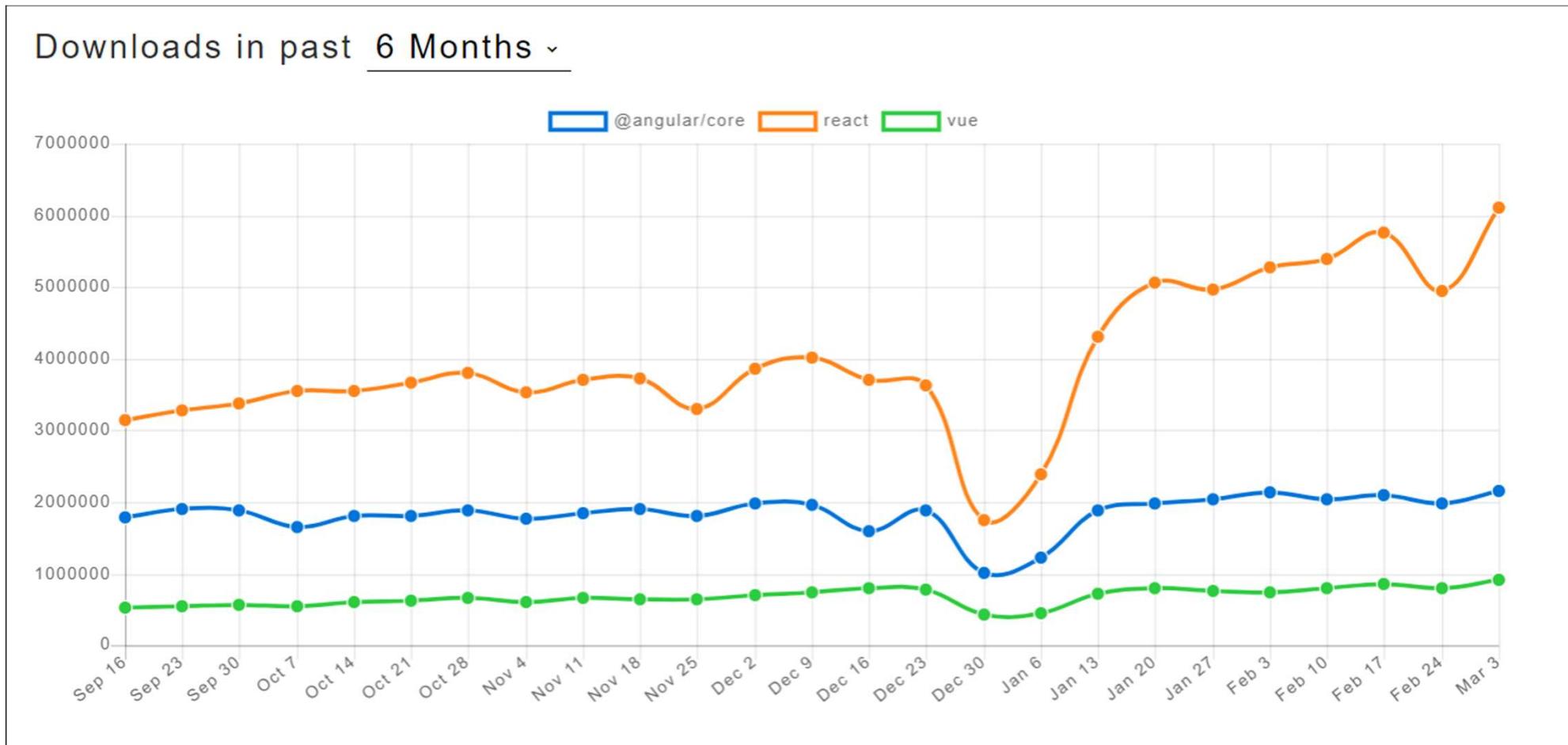
Backer
\$10 or more per month
Your name will be put in backers.md in the Vue.js repository.

JOIN \$10 TIER

Generous Backer
\$50 or more per month
Your name will be put at the top of backers.md in the Vue.js repository.

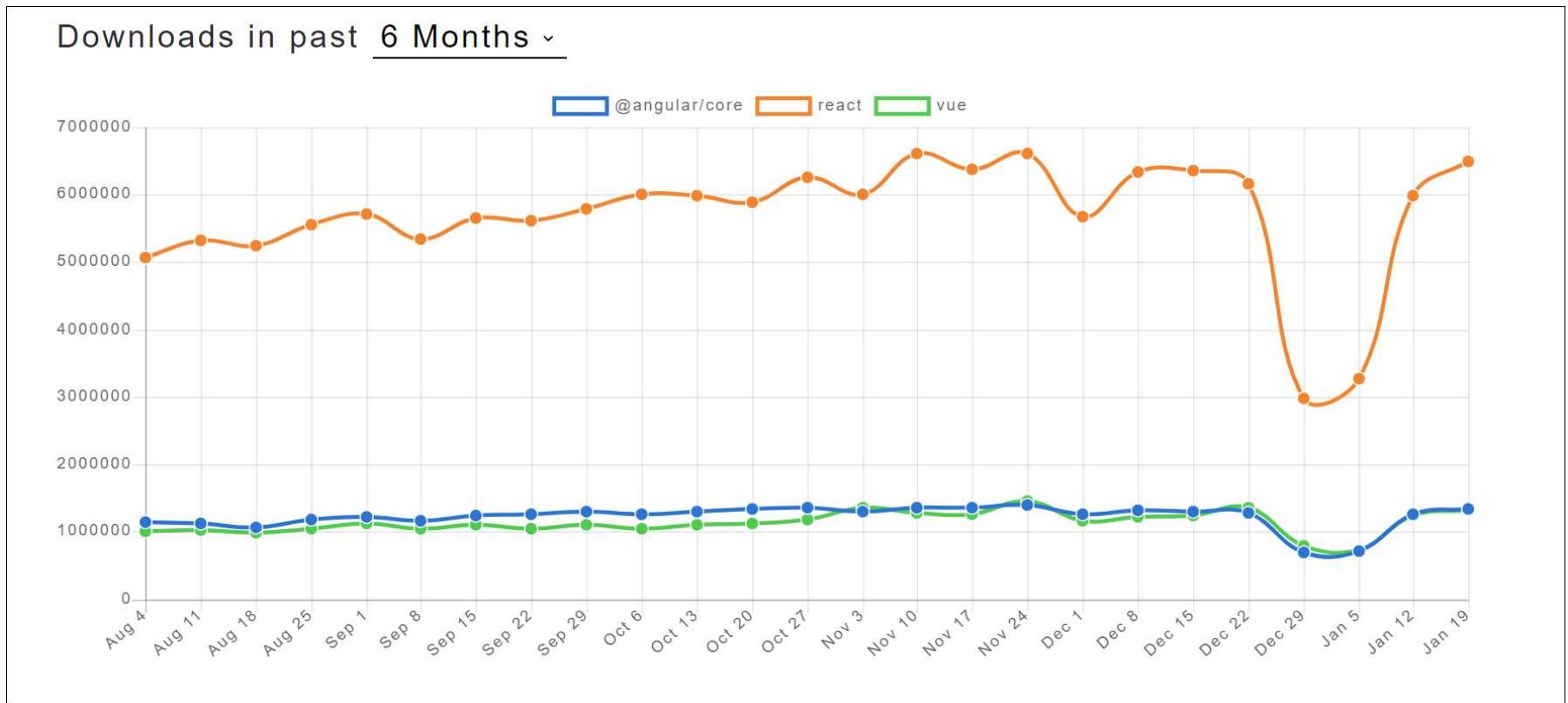
No backing from company with deep pockets – all crowd funded

A 2019 comparison



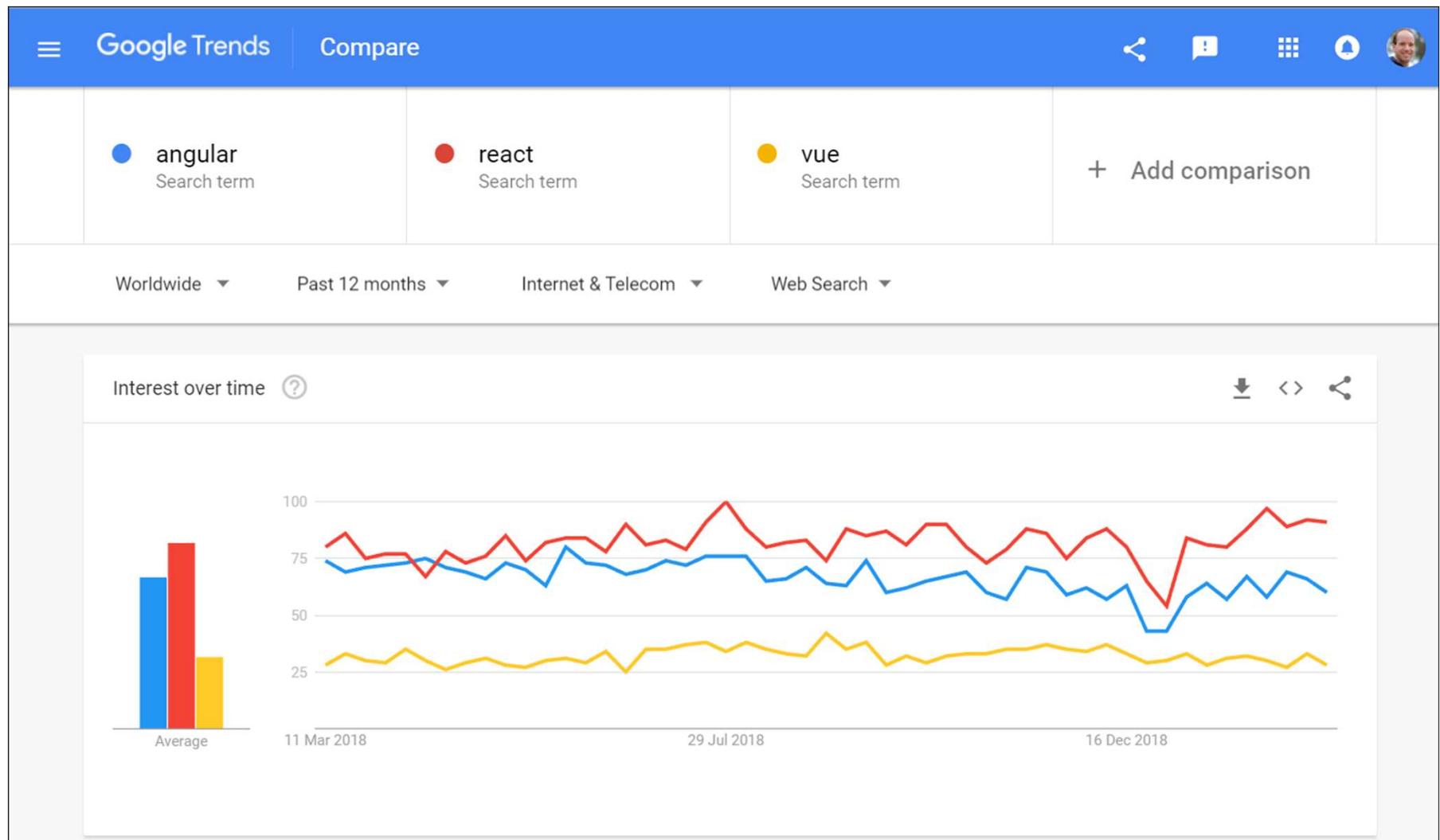
<https://www.npmtrends.com/@angular/core-vs-react-vs-vue>

A 2020 comparison



<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 ...

| | |
|---|----------|
| 1. freeCodeCamp/freeCodeCamp | ★ 298396 |
| 2. vuejs/vue | ★ 129161 |
| 3. twbs/bootstrap | ★ 128833 |
| 4. facebook/react | ★ 124937 |
| 5. tensorflow/tensorflow | ★ 123450 |
| 6. EbookFoundation/free-programming-books | ★ 118765 |
| 7. sindresorhus/awesome | ★ 99459 |
| 8. d3/d3 | ★ 83265 |
| 51. zeit/next.js | ★ 35904 |
| 52. moment/moment | ★ 35598 |
| 53. nodejs/node-v0.x-archive | ★ 35577 |
| 54. ionic-team/ionic | ★ 35555 |
| 55. typicode/json-server | ★ 35389 |
| 56. mrdoob/three.js | ★ 35013 |
| 57. rust-lang/rust | ★ 34749 |
| 58. 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

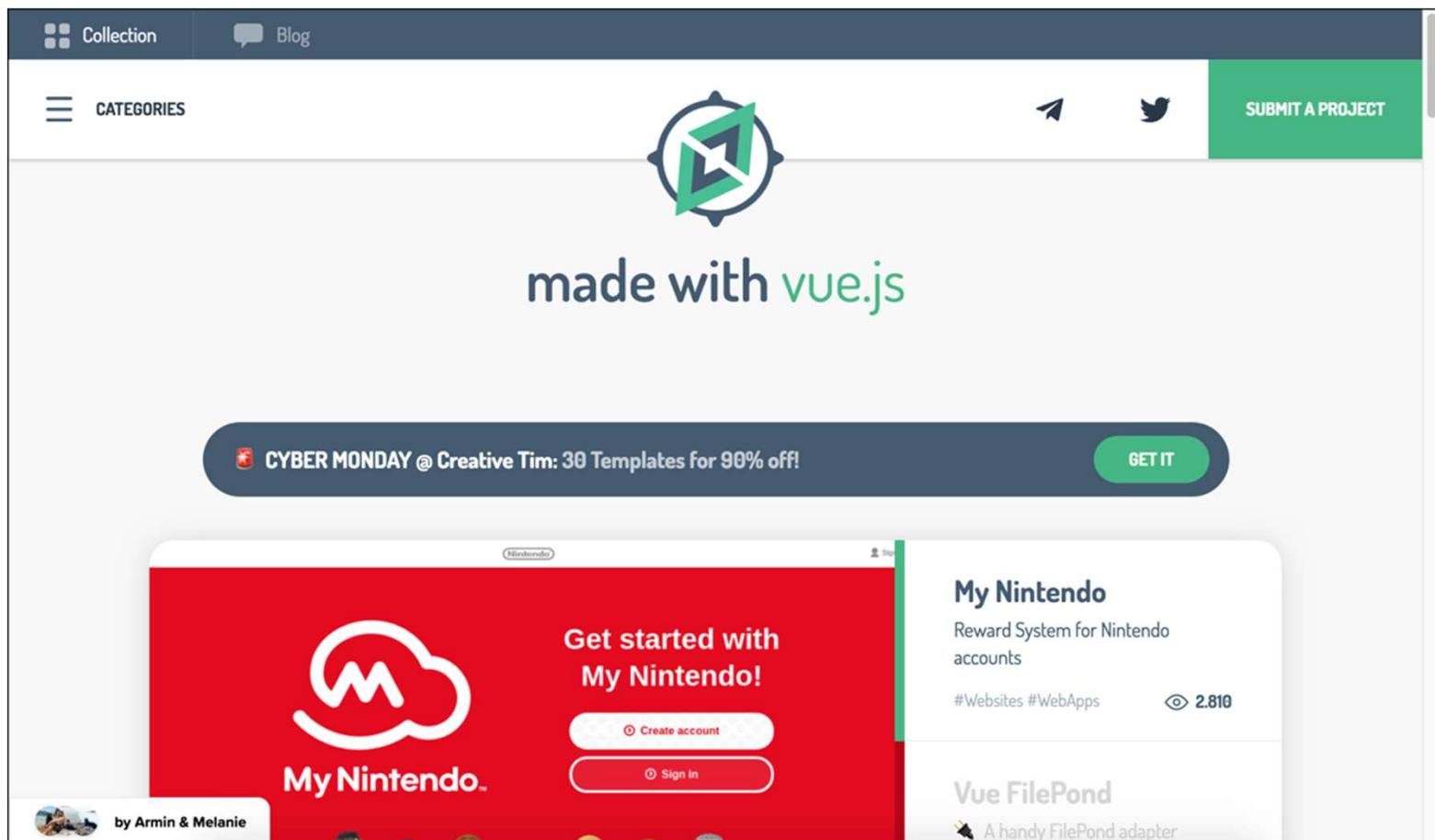
38



[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?

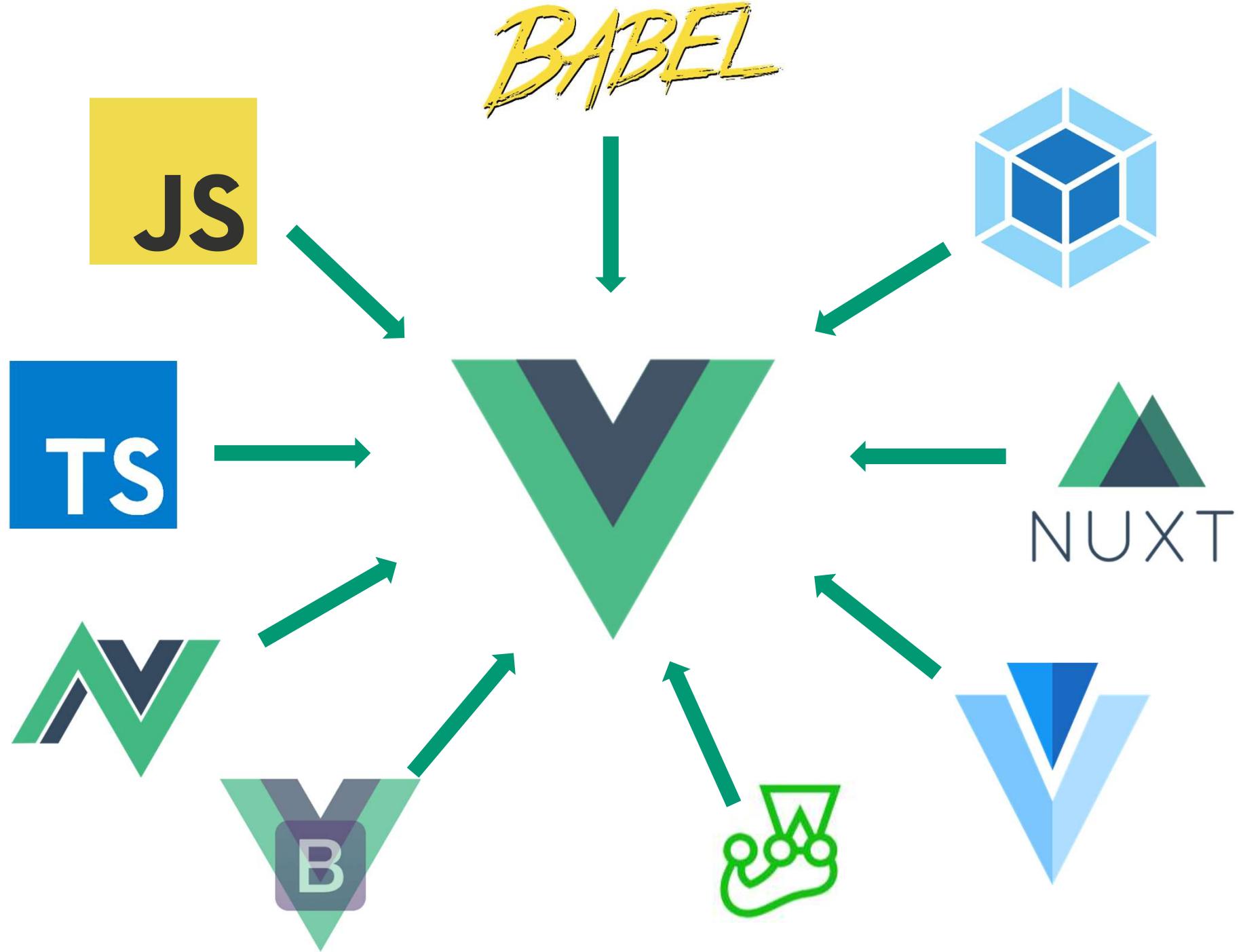
Hoe voorkom ik phishing?

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

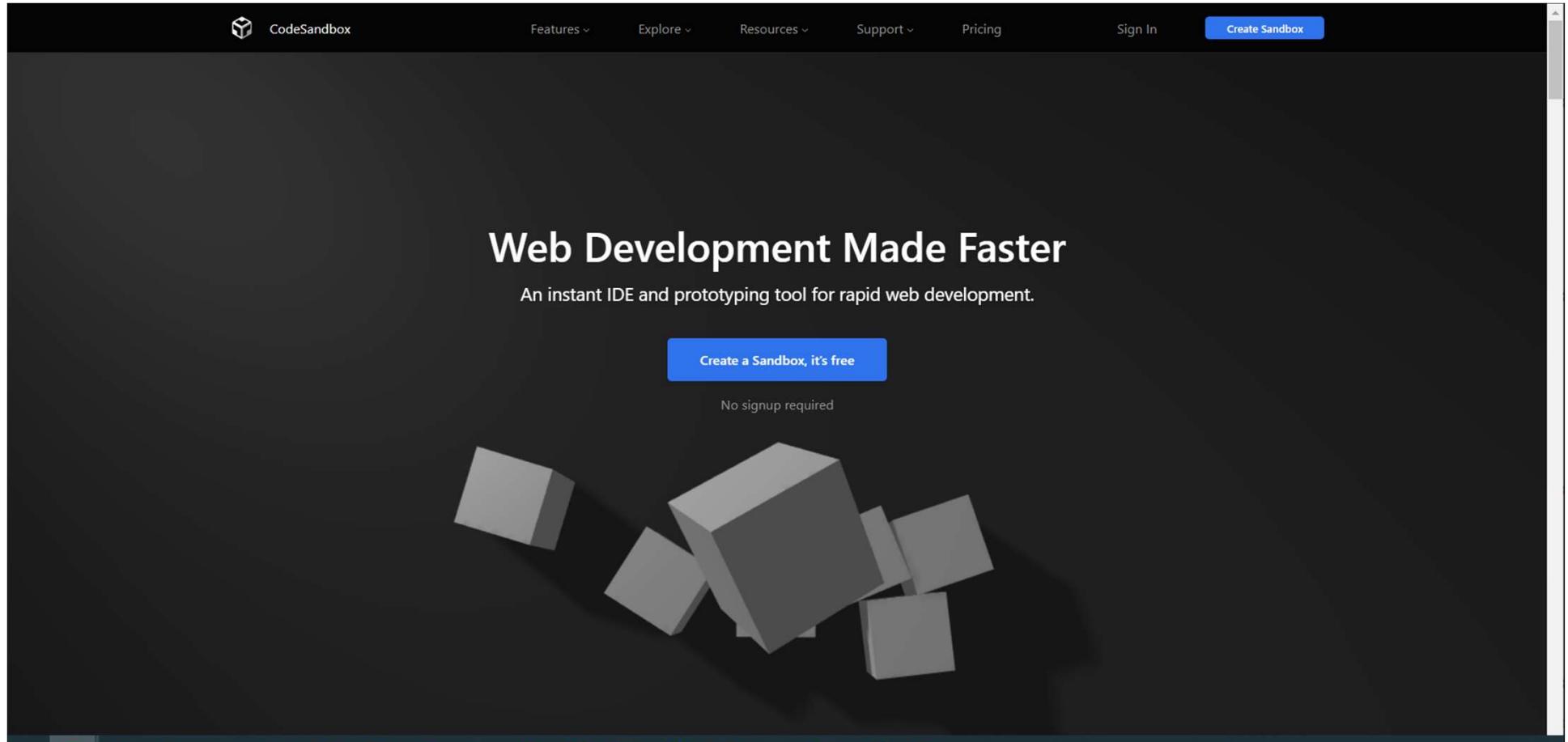


Additional services, options, etc.

The image displays four separate screenshots of web-based tools, each with a distinct color scheme and layout:

- Amplify Framework:** A white header bar with the "Amplify Framework" logo and navigation links for "Get Started", "Toolchain", and "St". Below this is a large orange section containing the text "The foundation for your cloud-powered mobile & web apps" and a "GET STARTED" button. To the right of the button are icons for Apple, Android, a globe, and React.
- Cloudinary:** A dark-themed header bar with "Cloudinary" and navigation links for "About", "Blog", and "Contact". Below this is a dark banner with "SOLUTIONS", "PRICING", and "CLOUDINARY" text.
- Vue Mastery:** A light blue header bar with the "Vue Mastery" logo and navigation links for "Courses" and "Pricing". Below this is a large blue section containing the text "The ultimate guide for Vue".
- CodeSandbox:** A dark-themed header bar with "CodeSandbox" and navigation links for "About", "Blog", and "Contact". Below this is a dark banner with "The online code editor for Web". At the bottom are two buttons: "OPEN EDITOR" and "EXPLORE EXAMPLES".

Codesandbox.io



<https://codesandbox.io/>

Online prototyping and development tool – develop in the cloud, no installation necessary

Features

| Features | Community | Pro |
|---|-----------|-----|
| Development & Prototyping | | |
| Templates Start from an official template, or create your own. | ✓ | ✓ |
| Static File Hosting All static files served via CDN. | ✓ | ✓ |
| Public Sandboxes Sandboxes, both the app and code, are available publicly by default. | ∞ | ∞ |
| Unlimited Private Sandboxes Set a sandbox as private or unlisted so others can't see the code or the app. | | ✓ |
| Private GitHub Repos Import and sync repos which are private on GitHub to CodeSandbox. | | ✓ |
| Online IDE | | |
| VS Code powered Editor Leverage the power and familiarity of VS Code. | ✓ | ✓ |

And more. . . .



On Vue.js versions

Vue.js is evolving

Current version – Vue 2.x

Vue.js Learn ▾ Ecosystem ▾ Team Resources ▾ Support Vue ▾ Translations ▾



The Progressive JavaScript Framework

 WHY VUE.JS? GET STARTED  GITHUB

Special Sponsor

Build APIs you need in minutes instead of days, for free.

| Approachable | Versatile | Performant |
|---|---|---|
| Already know HTML, CSS and JavaScript? Read the guide and start building things in no time! | An incrementally adoptable ecosystem that scales between a library and a full-featured framework. | 20KB min+gzip Runtime Blazing Fast Virtual DOM Minimal Optimization Efforts |

Waiting for images.opencollective.com...

Around the corner – Vue 3

New features

Better performance

Backward compatible

New features are *opt-in*



Stage: Release Candidate (RC), as of

September 2020

Evan You on Vue 3

The screenshot shows a YouTube video player. The video title is "What's coming in Vue 3.0" by "evan you vue 3". The video content features a man speaking at a podium with a laptop, and a slide with the text "What's coming in Vue 3.0" and a bulleted list of five items. The list is: "● Make it faster", "● Make it smaller", "● Make it more maintainable", "● Make it easier to target native", and "● Make your life easier". The video has a timestamp of 0:43 / 39:20. The YouTube interface includes a sidebar with sponsor logos like Rangle.io, ChainDigit, bugsnag, ionic, netlify, Cloudinary, Snipcart, VueJobs, Tighten, and a "Vue.js Crash Course" thumbnail.

What's coming in Vue 3.0

- Make it faster
- Make it smaller
- Make it more maintainable
- Make it easier to target native
- Make your life easier

Evan You on what's coming in Vue 3.0

79.653 weergaven • 17 jan. 2019

Volgende

AUTOPLAY

Vue JS Crash Course

Traversy Media

846K weergaven • 1 jaar geleden

<https://www.youtube.com/watch?v=8Hgt9HYaCDA>

More info/summary on Vue 3

02.09.2020

Vue 3 – A roundup of infos about the new version of Vue.js

f t

By now you're probably aware we're getting a shiny new version of Vue. Evan you announced it back in Fall 2018 at Vue.js London.

Until the release of Vue 3, we're going to collect all valuable infos and resources in this article for you, so you'll be up-to-date.

 by Armin & Melanie

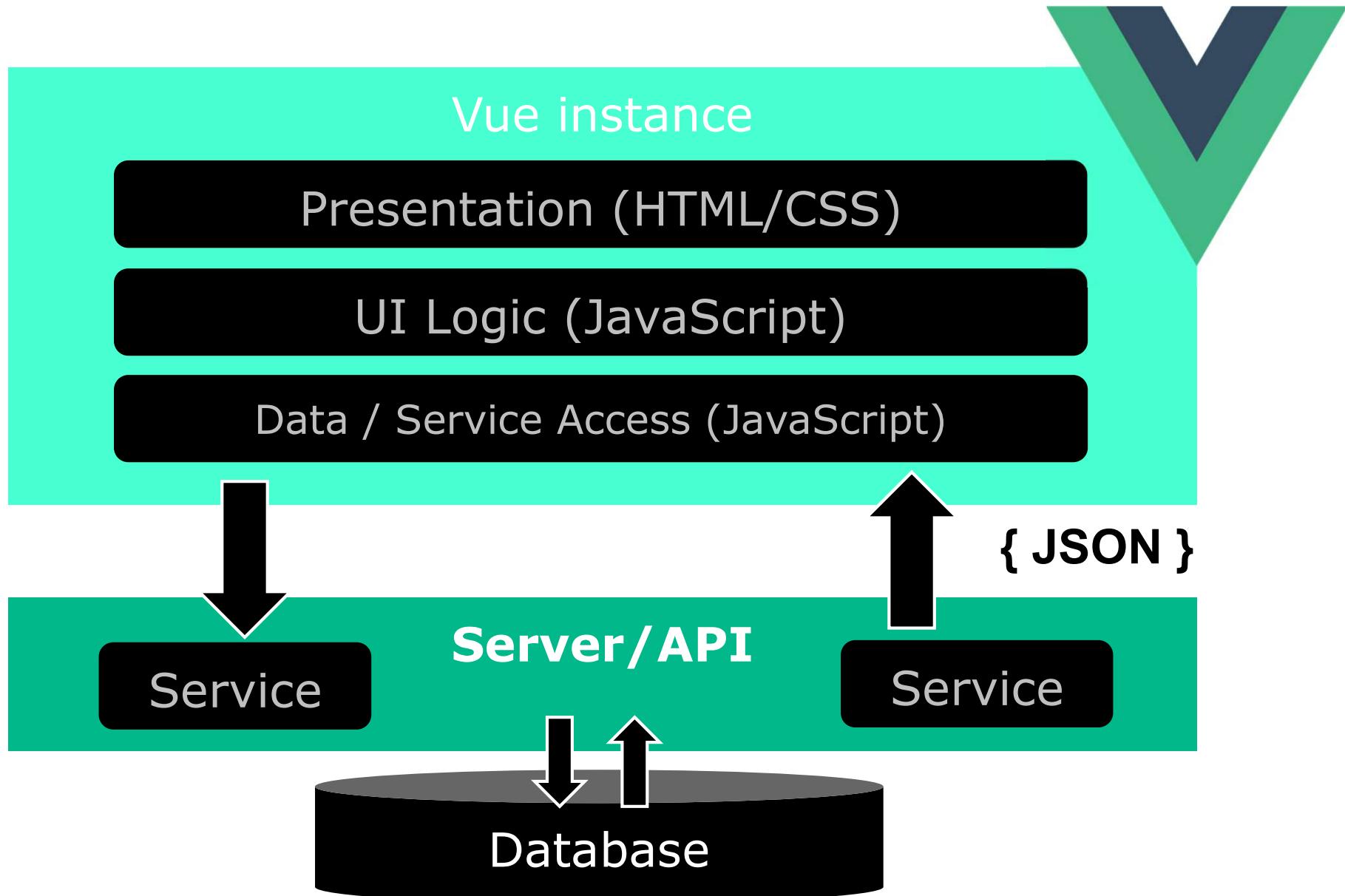
<https://madewithvuejs.com/blog/vue-3-roundup>



Vue application structure

Getting a conceptual understanding of the Vue app architecture

Single Page Application

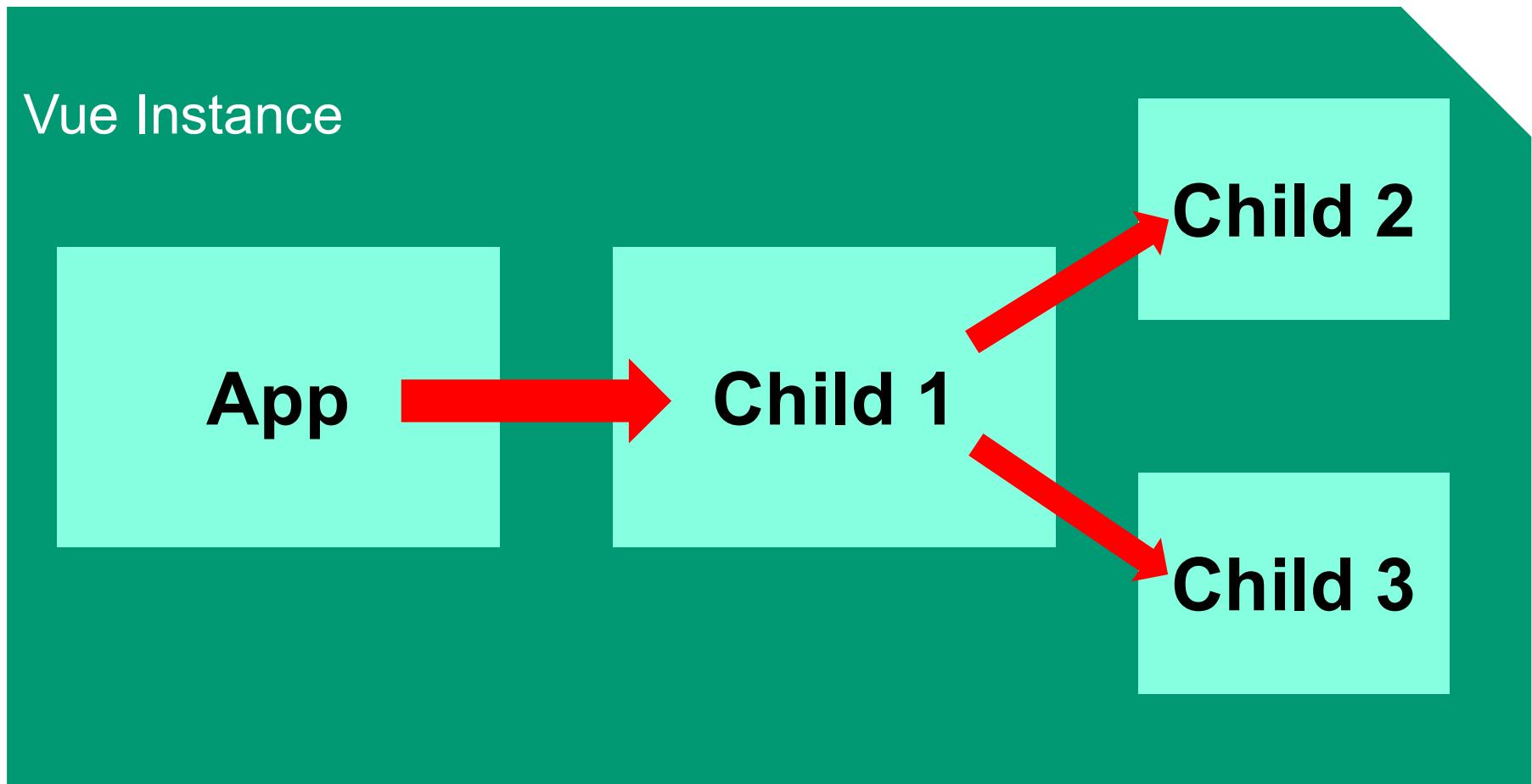


The Vue instance

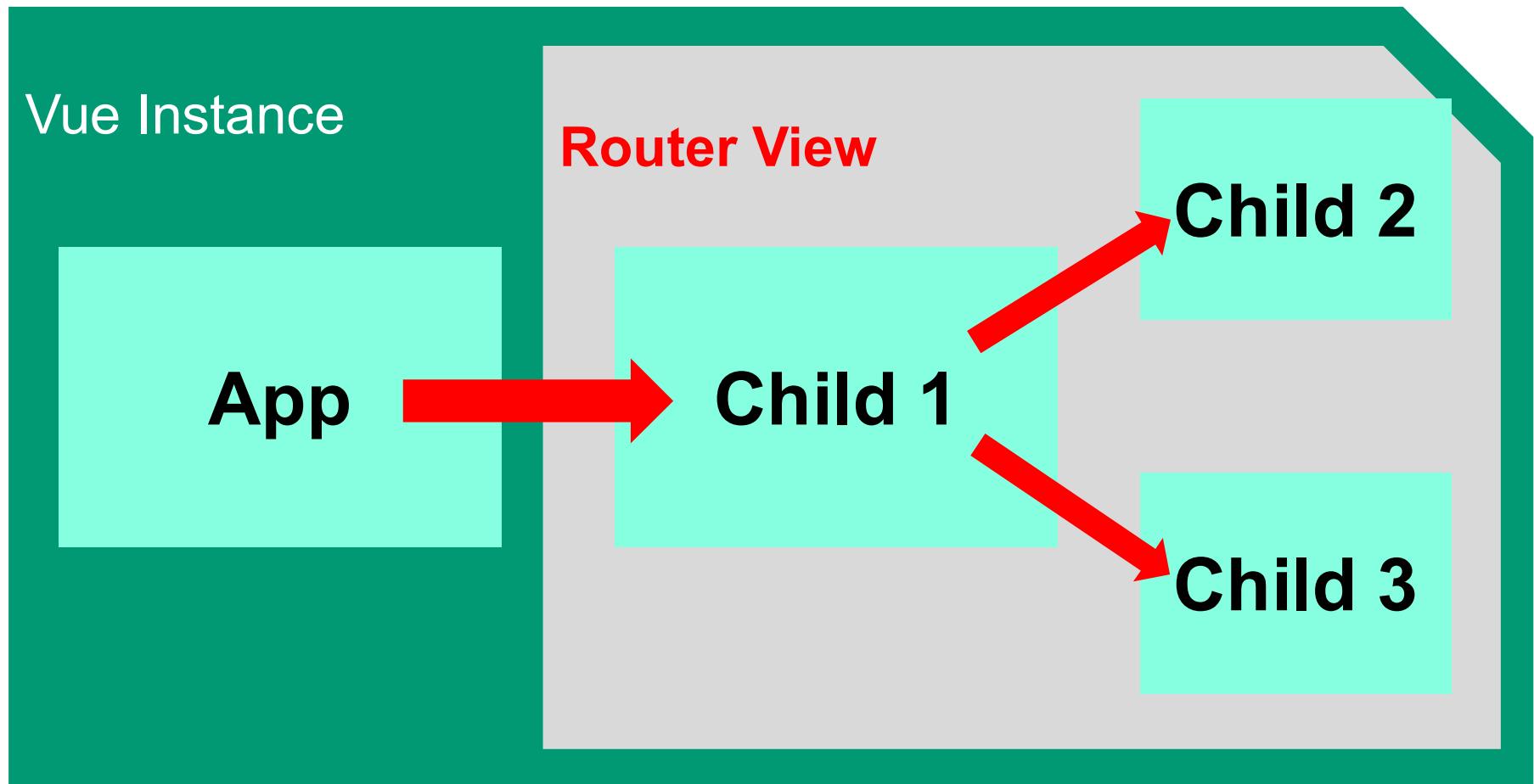
- The heart of every application : the **Vue instance**
- Created whenever you call `new Vue()` in your code
- Typical application: just 1 (one) Vue instance
 - File: `main.js`
- Pass in a configuration object
- Render a component. Typically by calling

```
new Vue({render : h => h(App)})
```

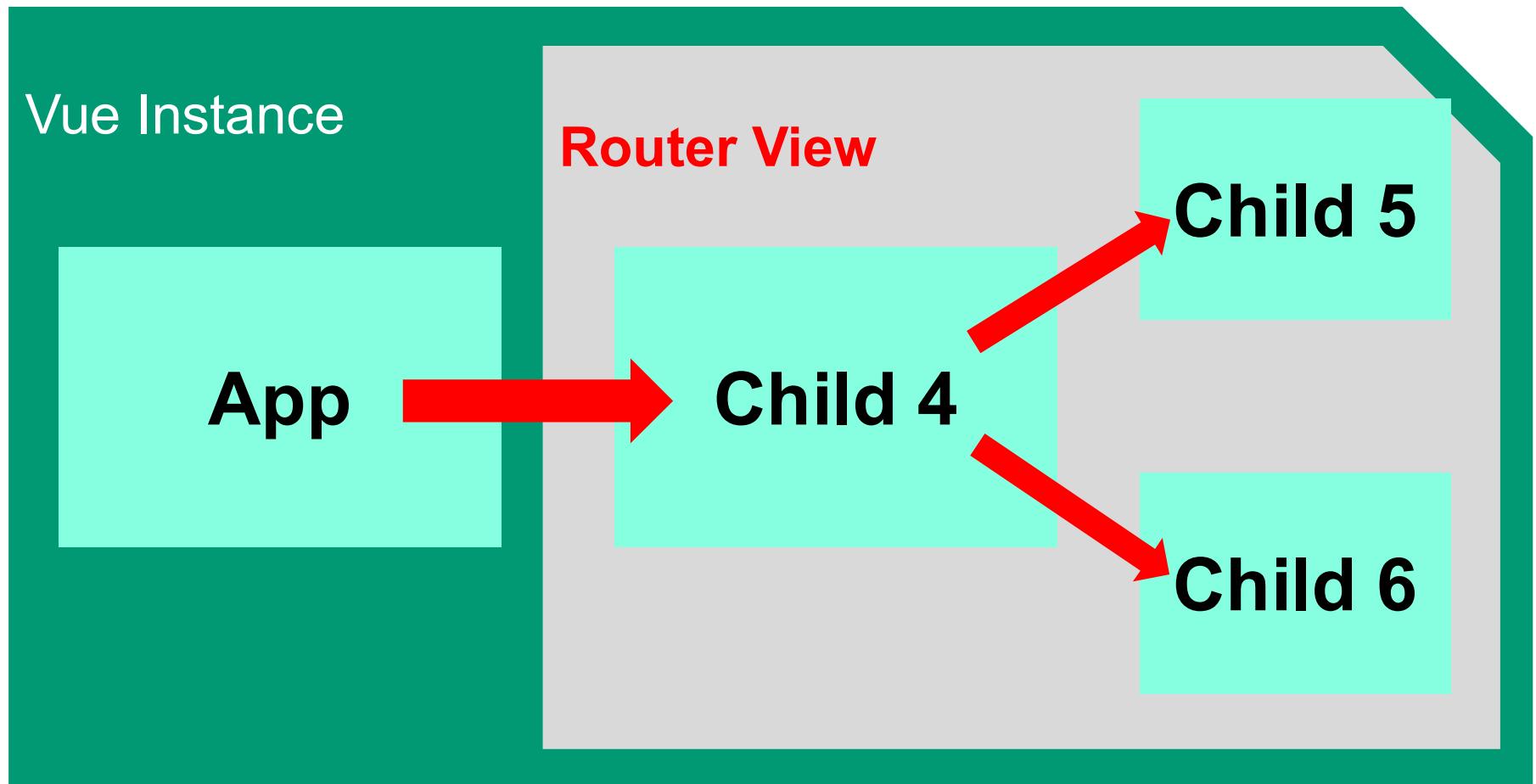
Visual of the Vue instance



Vue app with routing



As the URL changes, content gets updated





Let's create an app

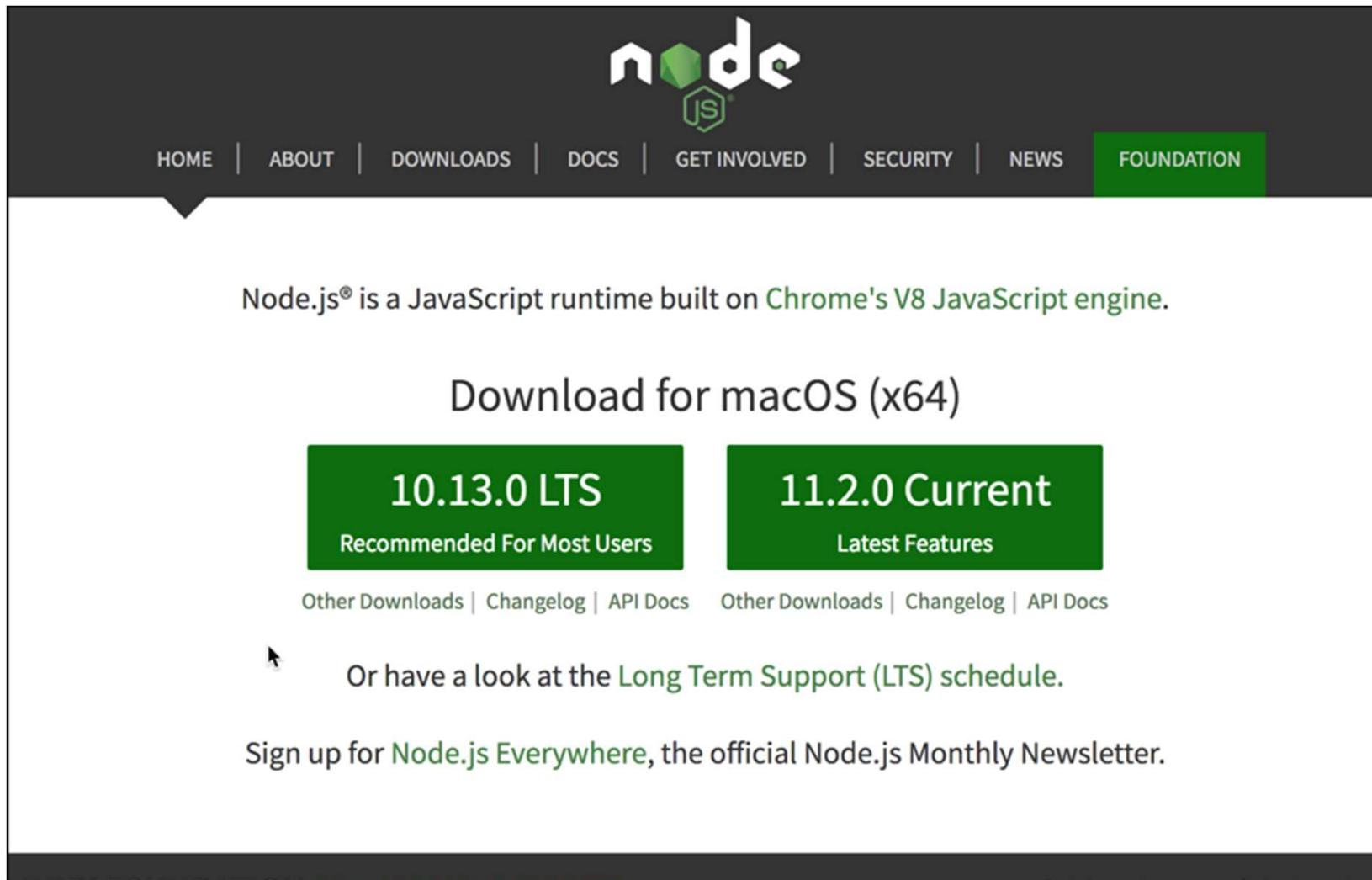
Install tooling and create your first app

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 page features a search bar at the top right. On the left, there's a sidebar with links for About, Documentation, Downloads (selected), GUI Clients, Logos, and Community. A note about the 'Pro Git book' is also present. The main content area is titled 'Downloads' and includes sections for Mac OS X, Windows, and Linux/Unix. A large image of a Mac monitor displays the 'Latest source Release 2.19.2' with a 'Download 2.19.0 for Mac' button. Below this, a note says 'Older releases are available and the Git source repository is on GitHub.' At the bottom, a terminal window shows a user named 'PeterKassenaar' running the command 'git --version', which outputs 'git version 2.17.2 (Apple Git-113)'. The output for this command is circled in red.

NodeJS



The screenshot shows the official Node.js website. At the top, there's a dark header bar with the Node.js logo and a navigation menu. The menu items are: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. The FOUNDATION item is highlighted with a green background. Below the header, a large text area states: "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." Underneath, there are two prominent download buttons for macOS (x64): one for "10.13.0 LTS" (Recommended For Most Users) and another for "11.2.0 Current" (Latest Features). Both buttons have links to "Other Downloads", "Changelog", and "API Docs". A mouse cursor is hovering over the LTS button. Below these buttons, there's a link to the "Long Term Support (LTS) schedule". Further down, there's a call to action: "Sign up for Node.js Everywhere, the official Node.js Monthly Newsletter." At the very bottom of the page, there's a footer bar with some small text and icons.

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for macOS (x64)

10.13.0 LTS
Recommended For Most Users

11.2.0 Current
Latest Features

Other Downloads | Changelog | API Docs Other Downloads | Changelog | API Docs

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter.

<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), Config Reference, Plugin Dev Guide, Plugins, Changelog, Languages, and GitHub. A search bar is also present. On the left, a sidebar lists categories like Overview, 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' and a message 'This project is sponsored by bit'. A yellow warning box states: 'WARNING This documentation is for `@vue/cli` version 3.x. For the old `vue-cli`, see [here](#).'. Below this, it says: 'Vue CLI is a full system for rapid Vue.js development, providing:' followed by a bulleted list of features.

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/>

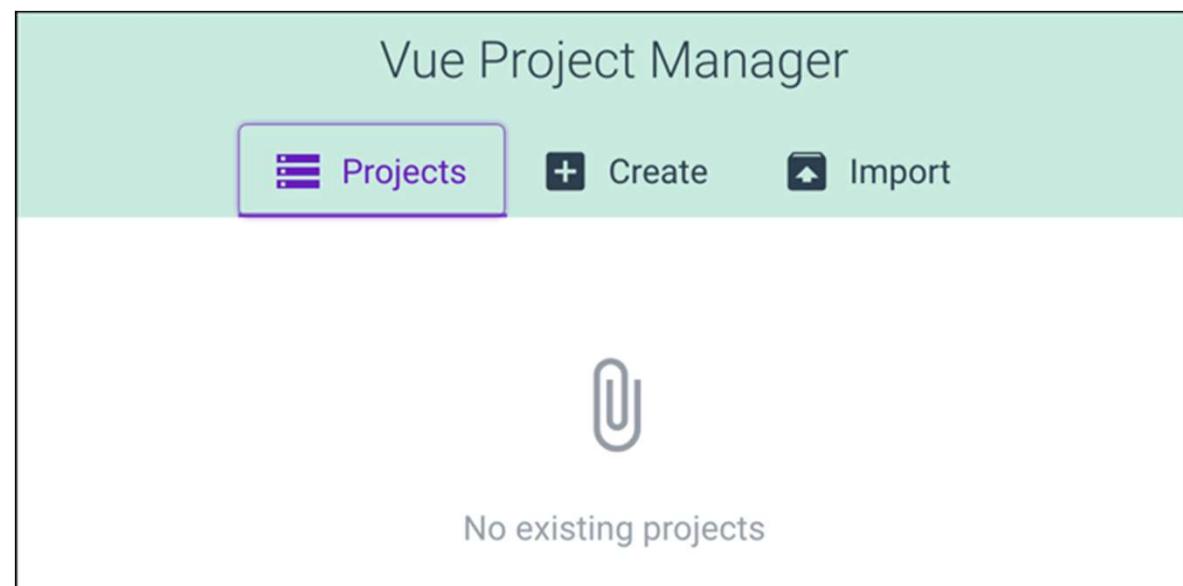
Installation of @vue/cli

```
[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`

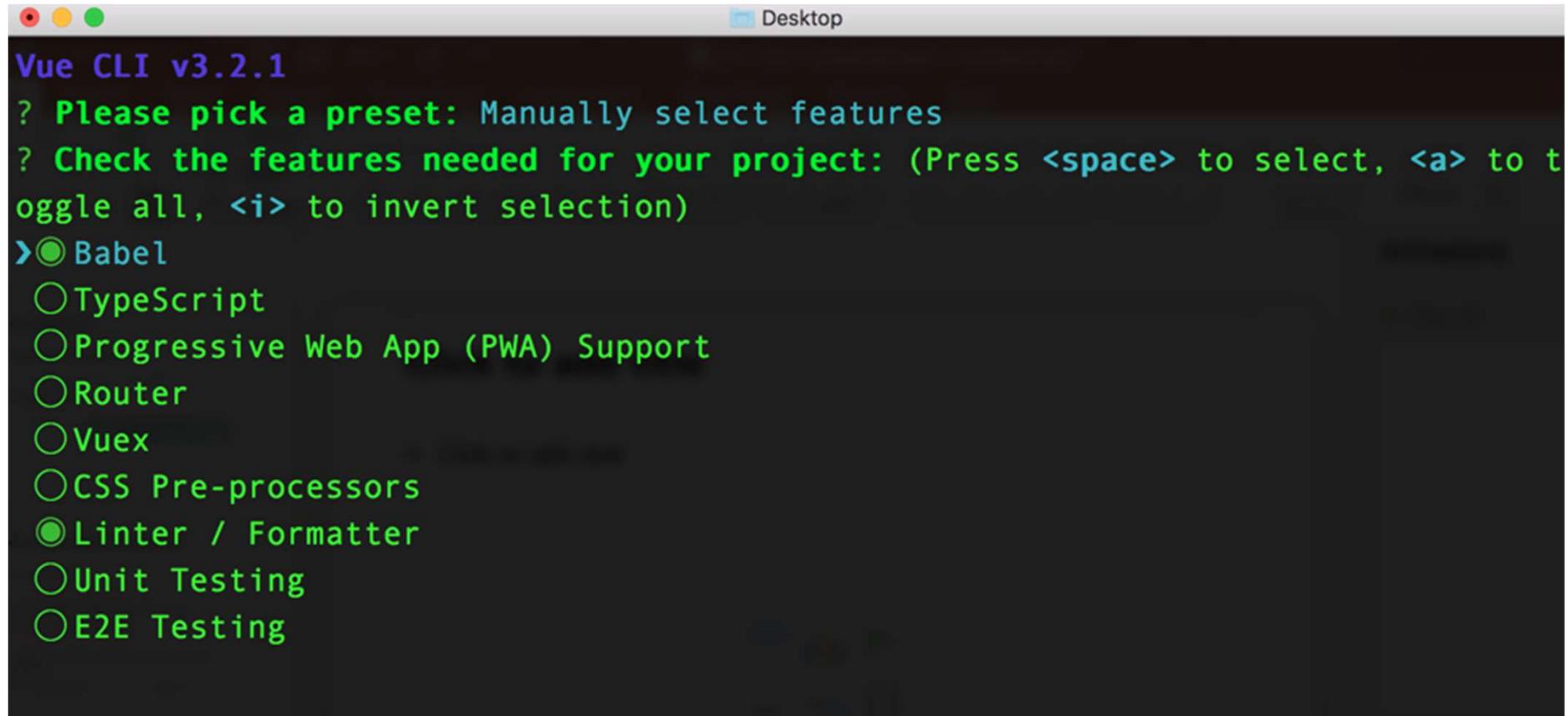
- Using the GUI
 - `vue ui`



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
```



A screenshot of a terminal window titled "Desktop". The title bar has red, yellow, and green close/minimize/maximize 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
```

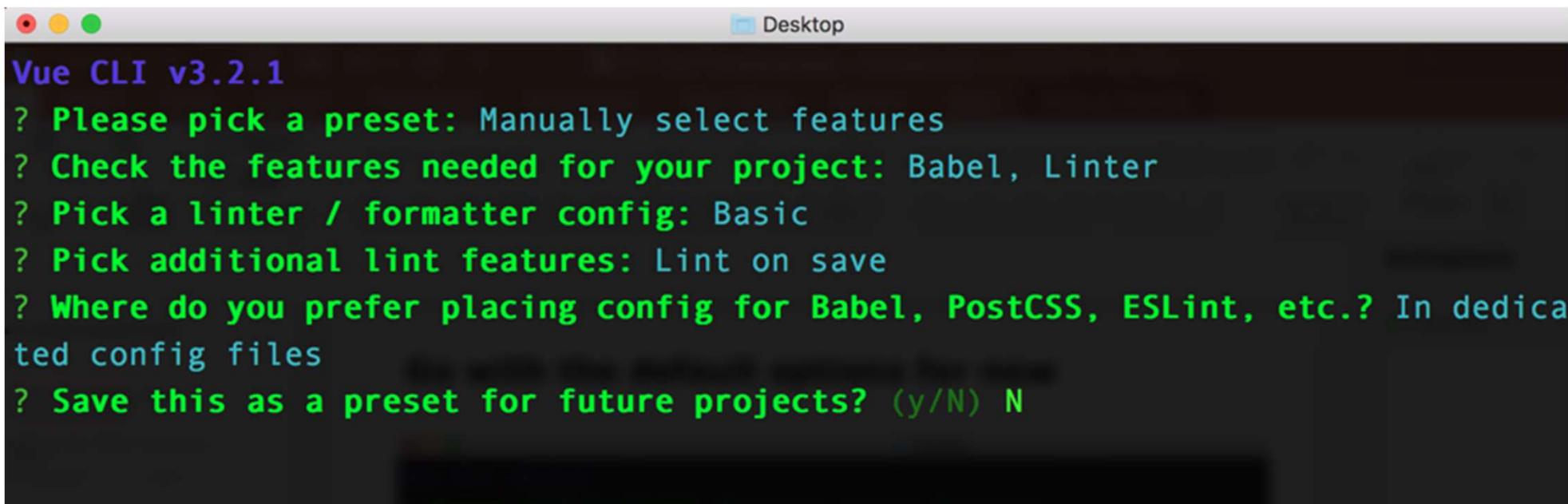
We're going to add Router and Vuex later

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
```



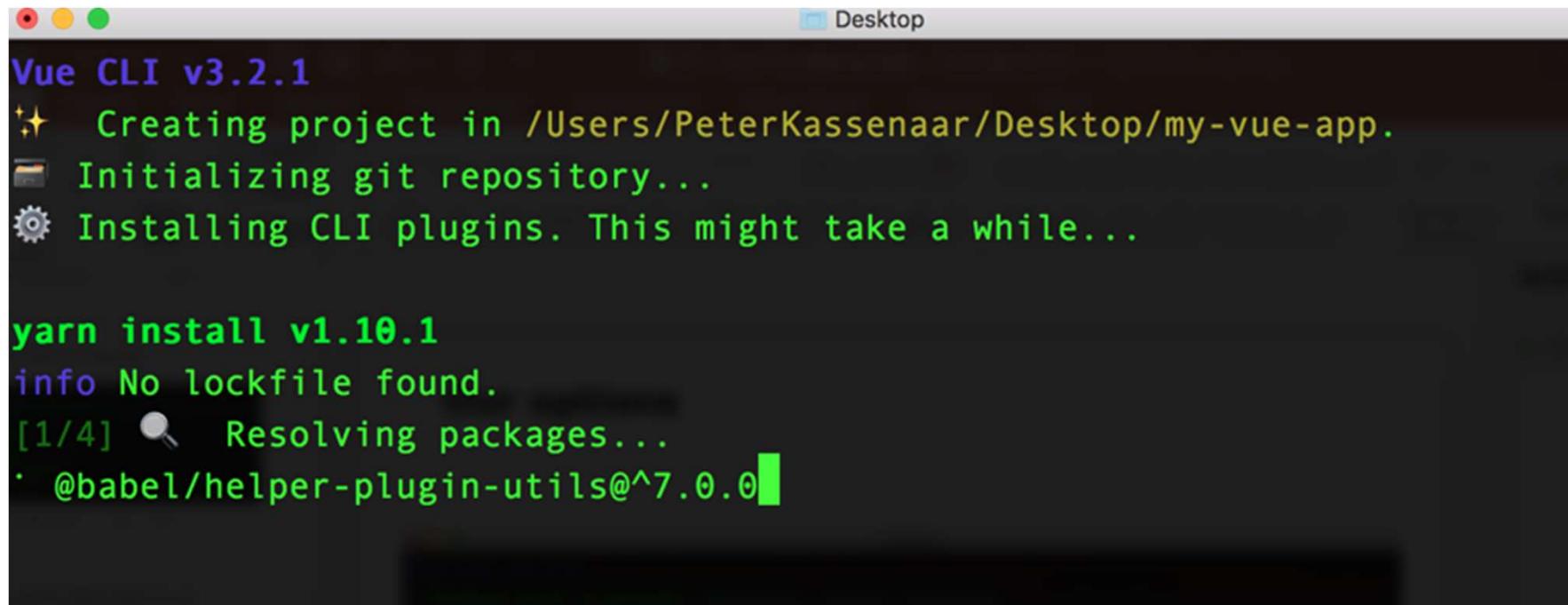
Our options



A screenshot of a terminal window titled "Desktop". The window shows the output of the Vue CLI v3.2.1 configuration process. The text is in white on a dark background. It starts with "Vue CLI v3.2.1" and then lists several configuration steps with question marks and descriptions. The last step is a confirmation question.

```
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...



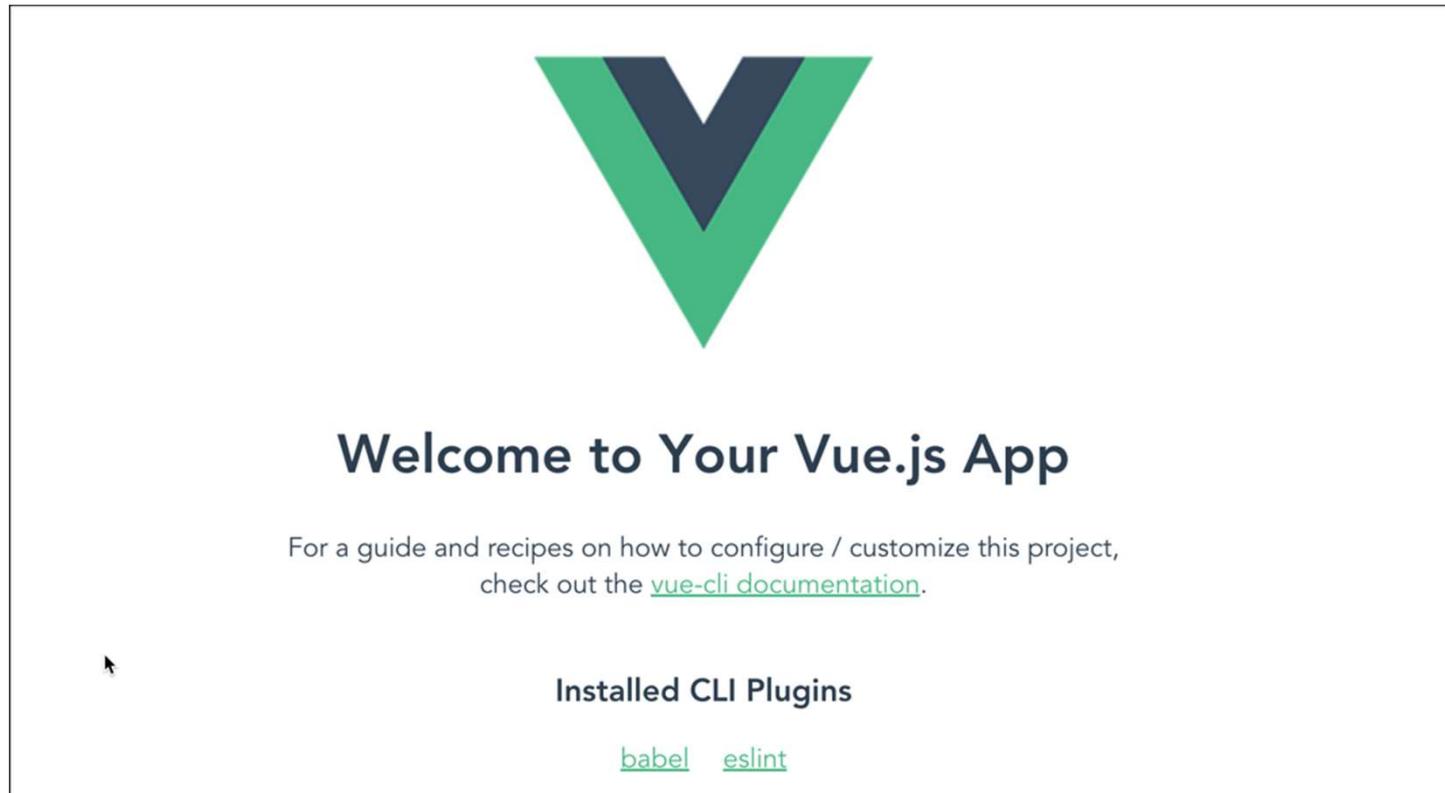
The screenshot shows a terminal window on a Mac OS X desktop. The title bar says "Desktop". The terminal output is as follows:

```
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
```

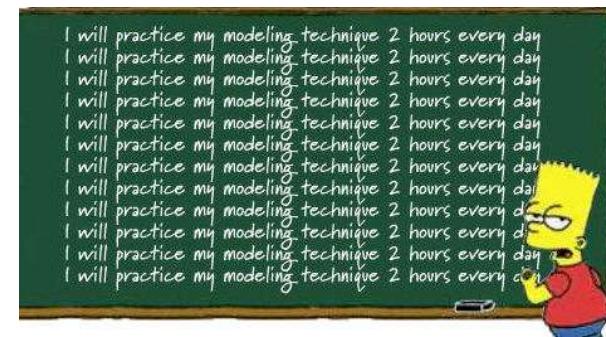
Standaardproject starten

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



Workshop

- Install `npm install -g @vue/cli`
 - <https://cli.vuejs.org/>
- Create a new, blank Vue project
 - `vue create my-awesome-app`
- Run it
 - `npm run serve`
- Replace the HTML in HelloWorld component with your own HTML
- Optional: use the GUI with `vue ui`
- Code example: [..../100-helloworld](#)



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"

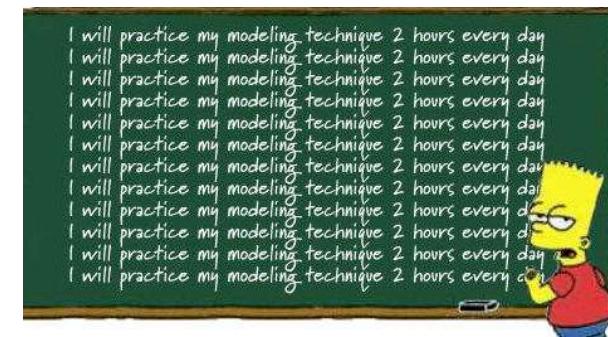
HTML

```
<!-- 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

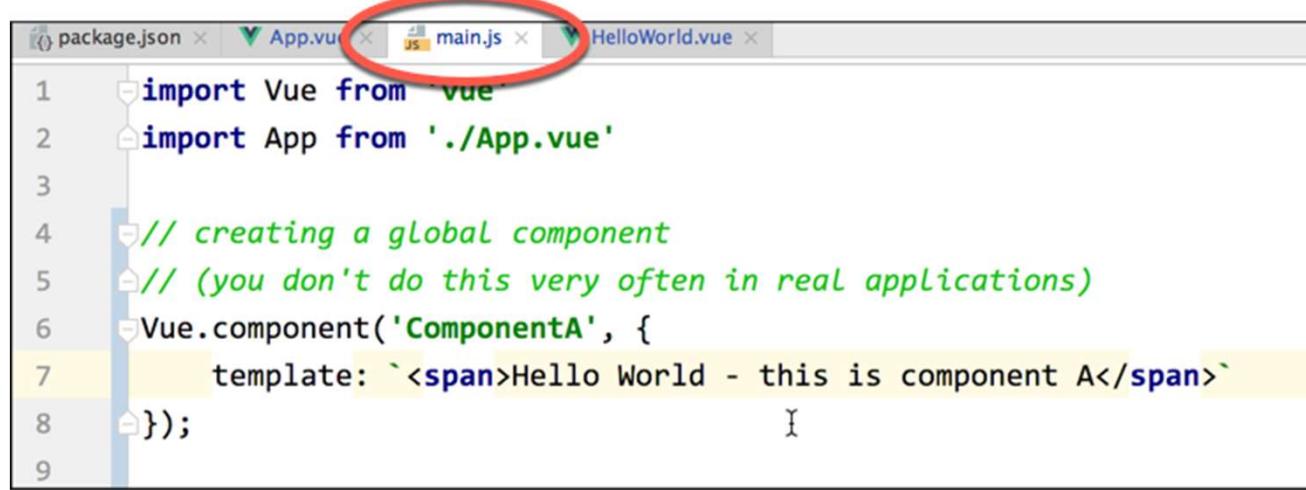
Adding and using your own components

Types of components

Global Components

**Single File
Components**

Global components



```
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
```



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

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

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

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 green icon. The code editor displays the following content:

```
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 first section, containing the `<template>` tag and its content, is highlighted.
- The second section, containing the `<script>` tag and its content, is highlighted.
- The third section, containing the `<style scoped>` tag and its content, is highlighted.

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 specific 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 in the script.
- An arrow points to the component entry in the components object of the export default block.

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 B: Strongly Recommended

Priority C: Recommended

Priority D: Use with Caution

Priority A Rules: Essential

Multi-word component names



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

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

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

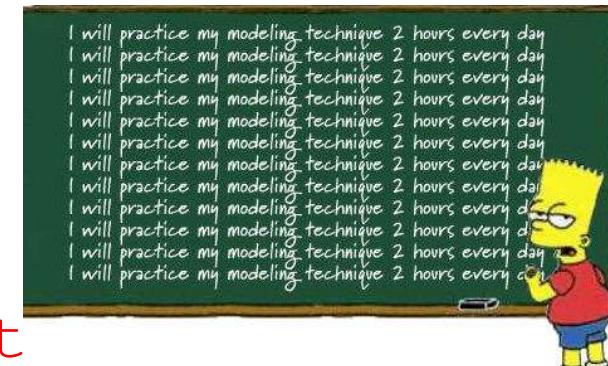
Self-closing components

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

<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