

Proprietors Assembly / Hausversammlung

Final Year Project

ALEXANDER GOGL

SUPERVISOR: CHRIS CASEY

COURSE: BSc (HONS) COMPUTING

University of Central Lancashire

19th February 2019

Contents

Introduction	ii
1 Literature Review	1
1.1 History of Vue.js	1
1.2 Comparison to other Frameworks	1
1.2.1 Vue 2.6 vs React 16	1
1.2.2 Vue 2.6 vs Angular 2	3
1.3 SEO for Single Page Applications	3
2 Implementation	4
3 Deployment	5
References	6
Literature	6
Online sources	6
List of figures	6
List of tables	7
Index	8

Introduction

Chapter 1

Literature Review

1.1 History of Vue.js

Vue was created by Evan You whilst working at Google on a UI-heavy project. Prior to the creation of Vue, there were no frameworks for rapidly prototyping UI interfaces. Subsequently, You, who did not find any viable solutions to his problem started working on his own. Angular, already widely used by developers and created by Google itself was too big and bloated of a framework to be sensible for small applications. React, a fairly new framework at the time also proved being too complex just like Backbone.js which was used for large-scale enterprise applications. None of these frameworks were adequate for prototyping [1, p. 10]. You's approach of reactive data-binding and reusable components later named Vue.js, was released in February 2014 [9] and helped fill that niche. Gradually improved over the years Vue is now utilized for complex enterprise-grade applications and small prototypes alike and has since been adopted by many developers and companies around the world. The most noteworthy of which are: Facebook, Netflix, Adobe, Xiaomi, Alibaba and GitLab [7]. With more than 130,000 stars on GitHub at the time of this writing, Vue is more popular than both React (122,000 stars) and Angular (45,000 stars).

1.2 Comparison to other Frameworks

1.2.1 Vue 2.6 vs React 16

Vue and React both are resemblant in terms of runtime performance, are based on a virtual DOM and are used for similar use cases. However, as with any framework or library there are slight differences which have to be taken into account prior to locking in to a particular technology. Both Vue and React can be used by integrating the respective library/framework into an existing site with a script tag or can be compiled with tools such as Webpack or Browserify. This section aims to provide a general overview, rather than discussing each form of integration in great detail.

Basic Usage

By default any valid HTML can be used to define Vue components, which is also referred to as a "template" in Vue terminology, whereas in React everything is defined in terms of JavaScript, meaning HTML often coupled with CSS are embedded into JSX render functions. Depending on the specific use case, on or the other provides advantages as well as disadvantages. JSX, which is a syntax extension to JavaScript [4] mixes the full power of a programming language with rendering and UI logic [4, 5], whereas templates are said to be easier to use as they are based on plain HTML. The following figure puts both concepts into contrast:

```

1  function getGreeting(user) {
2    if (user) {
3      return
4        <h1>
5          Hello, {formatName(user)}!
6        </h1>;
7    }
8    return
9    <h1> Hello, Stranger.
10   </h1>;
11 }

```

Listing 1.1: JSX Render Function

```

1  <template>
2    <div>
3      <h1 v-if="condition">
4        {{ user }}
5      </h1>
6      <h1 v-else>
7        "Hello Stranger"
8      </h1>
9    </div>
10 </template>

```

Listing 1.2: Vue Template

According to the Vue documentation JSX render functions provide:

- full leverage of JavaScript, including temporary variables and direct references to these
- and better tooling support than Vue (linting, type checking, autocompletion) [5].

On the contrary, Vue templates:

- are easier to use when coming from an HTML background,
- make it smoother to migrate existing applications to incorporate Vue,
- do not have a steep learning curve resulting in faster adoption by beginners
- and can be further enhanced with various preprocessors [5].

Component-Scoped CSS

Styling in React is most commonly done with CSS-in-JS solutions within JSX functions. In Vue applications on the other hand the styling of components is achieved by defining css classes within style tags, separating style and logic. By optionally using the scoped keyword in style tags, the css is bound to that specific component, reducing the risk of polluting global style sheets.

Scaling Applications

Applications built with either framework can both be easily scaled up or down depending on the applications requirements. For creating sophisticated and larger applications, three parts have to be incorporated in general: The core library, routing and state management. While all of these are officially provided by the Vue core team for Vue applications, React leaves routing and state management to the community [5]. Vue and React both offer an optional CLI generator interface to scaffold projects, however, Vue offers more options, leaving the choice to the developer which building system and plugins to use. React's more limiting approach with create-react-app makes it easier to start a project as it only needs a single dependency but limits the user to a given setup, which can however, be moved to a more customized environment with little effort.

If the goal is to add reactive and interactive elements to an existing webpage both frameworks can be used by adding script tags to any site [3, 5]. The obvious advantage is that no bundler has to be used in order for code to function but in doing so developers are deprived from being able to use plugins, preprocessors, and various other tools and are most commonly left with a larger bundle size.

1.2.2 Vue 2.6 vs Angular 2

Programming Language

Even though JavaScript could be used, Angular essentially requires the usage of TypeScript, a typed superset of JavaScript [8]. As opposed to JavaScript, TypeScript comes with static type checking which naturally makes applications less prone to errors [2] and therefore is often used within very large corporate projects. However, as with JSX, TypeScript is an additional learning step and can lead to decreased productivity in smaller projects. Being a dynamic language, JavaScript's ability to address quickly changing requirements makes it especially suitable for rapid prototyping [2, p. 72]. Nonetheless, Vue makes it possible to use TypeScript if the need arises [6].

Performance & Build Size

Both frameworks match one another in regards of performance but clearly diverge when it comes to build size. A typical Vue project with the additional dependencies of vuex (state-management) and vue-router uses up 30KB of space [5]. Angular applications built with the angular project scaffolding interface angular-cli are about twice as big, using up 65KB gzipped [5].

Flexibility & Learning Curve

Designed with the purpose of building large and complex applications Angulars API exposes a lot of functionality which is most commonly only necessary for exactly these types of applications. By providing pre-defined ways of interacting with the system, Angular inherently becomes opinionated and more difficult to master [5]. Vue gives developers the freedom to choose between various build systems and does not enforce a certain structuring of the application [5].

1.3 SEO for Single Page Applications

Chapter 2

Implementation

Chapter 3

Deployment with Docker

References

Literature

- [1] Olga Filipova. *Learning Vue.js 2*. Packt Publishing Ltd, 2016 (cit. on p. 1).
- [2] Francisco Ortin and Miguel Garcia. ‘A Programming Language That Combines the Benefits of Static and Dynamic Typing’. In: *Software and Data Technologies*. Ed. by José Cordeiro, Maria Virvou and Boris Shishkov. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 72–87 (cit. on p. 3).

Online sources

- [3] React contributors. *Add React to a Website – React*. URL: <https://reactjs.org/docs/add-react-to-a-website.html> (visited on) (cit. on p. 2).
- [4] React contributors. *Rendering Elements – React*. URL: <https://reactjs.org/docs/rendering-elements.html> (visited on) (cit. on p. 1).
- [5] Vue contributors. *Comparison with Other Frameworks — Vue.js*. 2018. URL: <https://vuejs.org/v2/guide/comparison.html> (visited on) (cit. on pp. 1–3).
- [6] Vue contributors. *TypeScript Support — Vue.js*. URL: <https://vuejs.org/v2/guide/typescript.html> (visited on) (cit. on p. 3).
- [7] Michał Sajnog. *13 Top Companies That Have Trusted Vue.js – Examples of Applications / Netguru Blog on Vue*. 02/18/2019. Mar. 2018. URL: <https://www.netguru.com/blog/13-top-companies-that-have-trusted-vue.js-examples-of-applications;%20http://gfs.sf.net/gerris.pdf> (cit. on p. 1).
- [8] *TypeScript - JavaScript that scales*. URL: <https://www.typescriptlang.org/> (visited on) (cit. on p. 3).
- [9] Wikipedia contributors. *Vue.js — Wikipedia, The Free Encyclopedia*. [Online; accessed 18-February-2019]. 2019. URL: <https://en.wikipedia.org/w/index.php?title=Vue.js&oldid=883864647> (cit. on p. 1).

List of Figures

List of Tables