

Frameworks comparison

I would like to introduce this chapter by clarifying that my practical knowledge web application frameworks is limited to the following 3: Angular (v4), Vue.js and bit of React. In the following sections I will briefly introduce each of the frameworks and do theoretical comparison of a specific framework to Vue. At the end I will perform practical comparison by implementing small project using all 3 frameworks.

Disclaimer: Content of this chapter is a based on my personal experience, knowledge and was enhanced with information from multiple web-pages and blog posts (~~see **Sources** section of this report for whole list~~).

Introduction, theoretical comparison

I will go over the frameworks beginning with the one the least similar to Vue (Angular), continue with React and end with Vue. Vue is the framework I worked the most with, because of that I am highly biased towards it, but I will do my best to not let this fact influence my reasoning.

Angular

Introduction

By Angular we understand Angular 2+ (Angular 1.X is known as AngularJS). It is an open-source web application framework that has been built by Google and released on September 2016. This powerful framework has been used for developing solutions for many global brands such as PayPal, Nike, Google, General Motors etc (TechMagic - Blog, 2019).

Project setup

Angular CLI³¹ is easy to install and use, it provides you with wide range of commands for creating project, generating components/routes/services, running local development server, executing official Angular linter, running tests etc.

Large scale solutions

There is a number of state management libraries for Angular, namely Akita, NGXS, and NGRX, which is the most popular and built upon Flux/Redux principles.

Syntax and architecture

I have spent my share of time working with Angular and to be honest that time was not spent pleasantly. Angular is known for its steep learning curve. The design and APIs are complex so getting up to speed with this technology, for someone who is new to it, is not exactly easy.

³¹ <https://cli.angular.io/>

It is highly recommended to use TypeScript with Angular as most of the learning resources and documentation are based on it. TypeScript as statically-typed language offers type checking, which is more than useful for highly scalable solutions. If the developer has C# or Java background, he might prefer to continue to use statically typed language. Although using Angular without TypeScript is possible, it is challenging and not advised. If you really want, you are of course able to use TypeScript within your Vue project as well, there are solutions that will help you to make things work. Angular uses two-way data binding³² (model state is changed when UI element is changed), but also offers one-way data binding, which emphasizes the idea of uni-directional data flow, that makes the flow easier to understand and debug. The same can be achieved in Vue as well.³³

Performance, UI update

From the performance point of view, both Angular and Vue are exceptionally fast and both of them support server-side rendering. Angular is known to be very large and although recently they managed to greatly reduce the size of the bundles, Vue solution, even with Vuex³⁴ and routing library, still wins this battle by a large margin.

Angular is relying on real DOM, the complete app is re-rendered after a change in a single component, unlike in Vue, which works with virtual DOM and only the components that should be updated are the ones that get updated, achieving better performance.

³² <https://www.pluralsight.com/guides/one-and-two-way-data-binding-angular>

³³ <https://medium.com/js-dojo/exploring-vue-js-reactive-two-way-data-binding-da533d0c4554>

³⁴ <https://vuex.vuejs.org/>

React

Introduction

React is a JavaScript library backed up by Facebook. It has been released at the end of May 2013. It quickly gained huge popularity as a more flexible alternative to Angular and it has been used to build apps like Yahoo! mail client, Khan Academy, Dropbox, Instagram etc. (Blog Brainhub.eu, 2019).

Project setup

When creating a new React project, you can use **create-react-app** npm package³⁵, but in comparison to **Vue CLI** it is very limited. Create-react-app does not allow any configuration during project creation, there is only 1 template. Limited setup customization could be a good thing, because if the application you strive to develop is simple, you don't have to worry about spending time setting things up.

Large scale solutions

For large scale applications both React and Vue offer solutions for state management (Redux/Flux for React, e.g. Vuex for Vue) and routing. Solutions for React are managed by the community so the ecosystem is more fragmented, but rich.

Syntax and architecture

In React everything is represented as JavaScript - HTML is represented by JSX³⁶ and recently the trend of putting CSS into JS grew in popularity. Having everything in JS allows you to use the full potential of it. It is worth to mention that React has steeper learning curve than Vue as it requires knowledge of JSX and ES6.

Performance, UI update

When it comes to runtime performance, both React and Vue are extremely fast. For updating the view after state has changed, React has to re-render component and its children (although the developer can fine tune the performance by explicitly disabling this behaviour for specific components). In my opinion in this area Vue does better job by keeping track on the dependencies automatically and thus knowing precisely which components should re-render based on the change in the state, because of that the developer does not have to worry about performance optimization too much and can focus on the development. Both React and Vue utilize virtual DOM^{37 38} to improve the performance of the UI.

³⁵ <https://github.com/facebook/create-react-app>

³⁶ <https://reactjs.org/docs/introducing-jsx.html>

³⁷ <https://programmingwithmosh.com/react/react-virtual-dom-explained/>

³⁸ <https://medium.com/@koheimikami/understanding-rendering-process-with-virtual-dom-in-vue-js-a6e602811782>

Vue

Introduction

Vue is a light-weight open-source JavaScript framework that has been heavily influenced by Angular and React. It has been initially released on February 2014 and is not backed by a large company. This extremely flexible framework has been used for project by multitude of companies such as Adobe, Alibaba, Wizz Air, GitLab and Grammarly (Netguru.com, 2019).

Project setup

During the project setup, Vue CLI allows you to choose from number of options for linting, CSS preprocessing (SASS, LESS, Stylus) etc. This allows you to generate new project that is tailored to your needs without having to spend sometimes even hours battling configuration.

Large scale solutions

It is certainly possible to use Vue application with Redux for state management, this idea lead to the creation of Vuex. Unlike Redux, which is developed and maintained by the community, Vuex is kept updated in alignment with the core Vue library - this and the fact that I have used Vuex a bit in my previous project, lead me to a decision to strengthen my knowledge of it by using it for this project as well.

Syntax and architecture

Vue is less opinionated than Angular and thus more flexible giving you freedom to do whatever you want and need. In contrast to React, Vue embraces the classic web technologies - any valid HTML is valid Vue template, which has number of advantages:

- Smaller learning curve, HTML is easier to work with for less experienced developers
- HTML is mature technology and for most of the developers it feels more natural to work with. Yes, you have to learn the special syntax Vue uses, but in my opinion it is quite simple
- Most of the existing web solutions use HTML so slowly migrating existing web solution into Vue to take advantage of its capabilities i