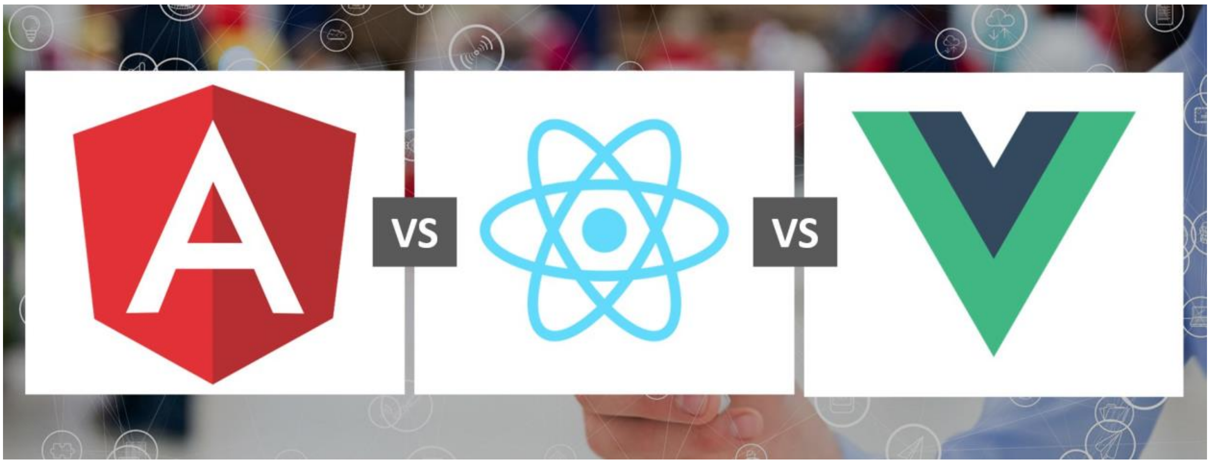
Angular vs React vs Vue



Deciding which JavaScript framework is best for your web application is never easy

In the last years there are tons of variant JavaScript frameworks emerged.

Angular, React, Vue are the most popular JavaScript frameworks at present and a comparison among these three library\framework is a challenging post to write.

Both these frameworks have their advantages and a huge fan base (especially Angular and React).

All three Frameworks are available under the ***MIT license*****.**

Before starting let's take a look at the current state and these technologies

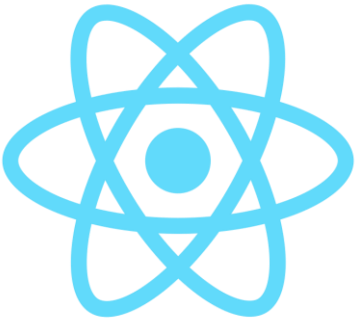
Angular: One framework.Mobile & desktop.

Angular is a TypeScript-based JavaScript framework. Developed and maintained by Google and was release in September 2016. Angular 2 is a **framework** to help us build client applications in HTML and JavaScript.

Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges.

The architecture diagram identifies the eight main building blocks of an Angular 2 application (Module, Component, Template, Metadata, Data Binding, Directive, Service, Dependency Injection)

Angular is used by Google, Wix, weather.com, healthcare.gov and Forbes



React: A JavaScript library for building user interfaces

React is a declarative, efficient, and flexible JavaScript library for building user interfaces.

Initially released in March 2013, was developed and is maintained by **Facebook.**

React is a ***JavaScript library*****. It's not a framework.**

React is often described as "the V in the MVC structure", in fact it's not a complete solution and we'll often need to use more libraries with React to form any solution. React does not assume anything about the other parts in any full solution. It focuses on just one thing, the View.

Vue: The Progressive JavaScript Framework

Vue is a **progressive framework** for building user interfaces.

Vue is one of the most rapidly growing JS frameworks in 2016, was released in February 2014 by ex-Google-employee Evan You and describes itself as a "Intuitive, Fast and Composable MVVM for building interactive interfaces."

The library is "**not a full-blown framework"** - it is designed to be a view layer that is simple and flexible

The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects.

Core development

The first difference among these three philosophies is that Angular is a **framework.**

It provides strong opinions as to how your application should be structured and also has more functionality out of the box.

React and Vue are **library** and are universally flexible, but this is not always a good thing.

In one project you must take a decision regarding its architecture or choose from different library and this can be really dangerous.

There are quite a lot of library for React and Vue on npm (Vue has fewer packages).

With Angular you don’t need to analyze libraries, you can just start working because it's comes with a confusing nest of build tools, boilerplate, etc...

Another difference between this library\framework is that Angular focuses entirely on **Single-Page-Applications (SPA)** and it's optimized for this type of solution, while React and Vue it's not focused on this type of application.

This does not mean that with React and Vue you can't create a SPA.There are some library that help us to create a SPA with these library (es react-router or vue-router), simply you can choose (another choose) between **Single-page-Applications and Multi-page-Application (MPA).**

All three libraries are component base, but focused on different language approach:

- React focused on **JavaScript ES6**
- Vue use **JavaScript ES5** or **ES6**
- Angular use **Typescript**

It's possible to write applications in React or Vue that use **Typescript** or **Flow** (a static type-checker developed by Facebook for JavaScript).

An important difference between this library it's the UI managements.

With React you use JSX**.**

JSX is an optional preprocessor for HTML-like syntax which will be compiled in JavaScript later.

With **JSX** UI templates and inline JavaScript logic are intermixed.

In Angular templates are enhanced HTML with special Angular language (*ngIf, ngFor, etc..) that allow to create a dynamic view.

With Angular you can create view with HTML , a technology that is simple and a standard in the web programming, but forces you to learn the Angular-specific syntax.

With Vue you have templates, scripts and styles are in one file in different section.

Vue define itself as **single-file components**.

And about the **separation of concerns?**

In angular we have a clear (but not so clear) separation of concerns for each components (.ts file for logic , HTML for View and Css file for style) while with Vue and React this is not true.

Supporter for React assert that separating templates and logic is merely a separation of technologies, not concerns.

Vue in his documentation about single page components , affirm that: "**separation of concerns is not equal to separation of file types". ** In modern UI development, we have found that instead of dividing the codebase into three huge layers that interweave with one another, it makes much more sense to divide them into loosely-coupled components and compose them. Inside a component, its template, logic and styles are inherently coupled, and collocating them actually makes the component more cohesive and maintainable."

Another big difference between this three library/framework is the management of **Binding**, between the view and data.

With Angular **two-way-binding** the model is updated when a change (ex. a change in inputs or select) in the view occurs.

With this way we have a cleaner and simple code to implements.

React only goes one way: it updates the model first and then it renders the UI element (**one-way-binding**).

One of the basic principle for React is the "Single source of Truth" where the state, that is read only, is store in a single place, then we can send that state down as props to your child components. Using this technique, when any part of that master state changes it will automatically update the props of your child components, and the changes will flow down in one direction from top to bottom – always synchronized, never duplicated.

This results in a better data overview, because the data only flows in one direction and makes debugging easier.

Vue allow you to choose between **two-way-binding** and **one-way-binding**, but the default is **one-way-binding.**

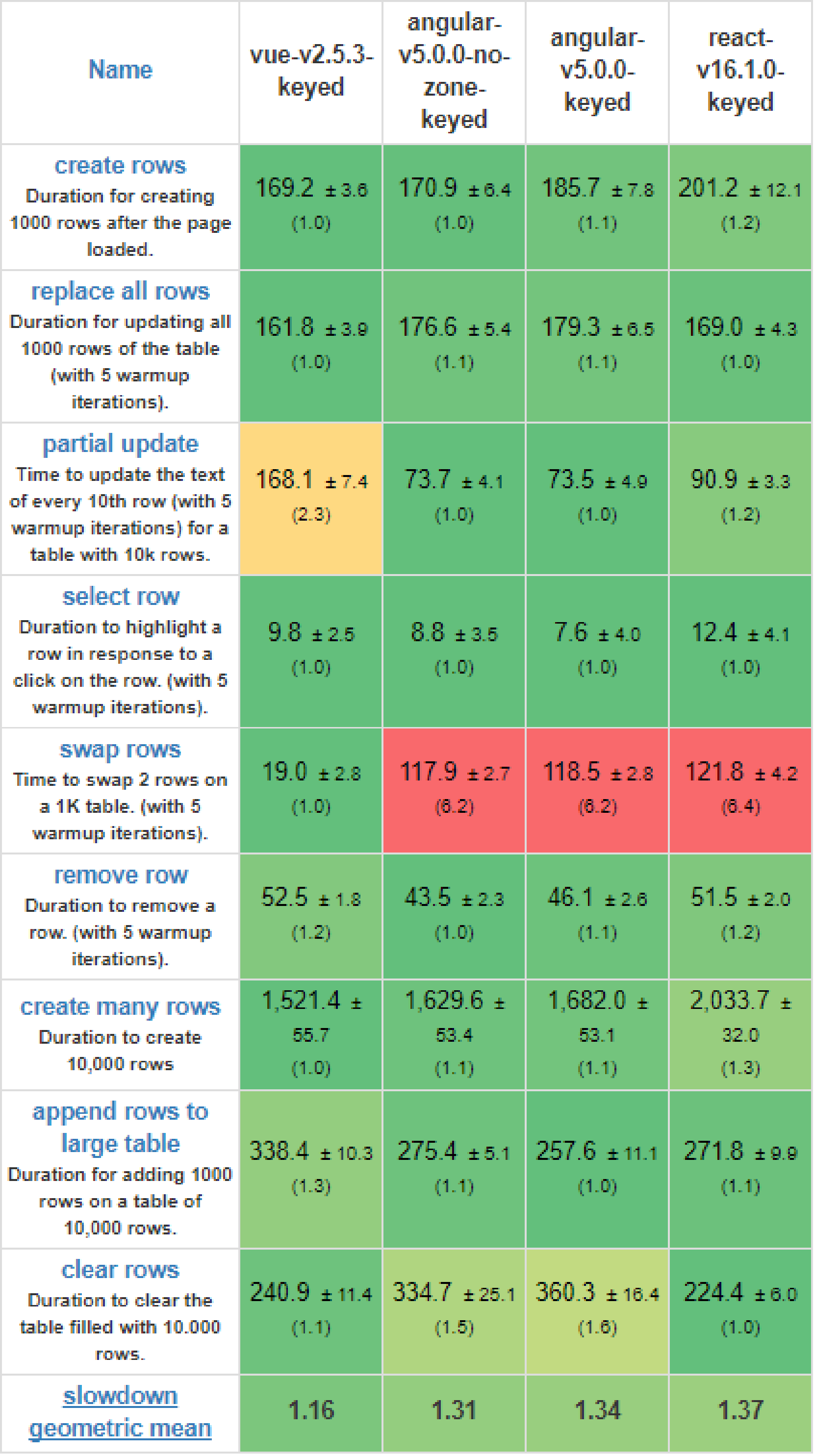
Coding And Performance

As mentioned previously Angular is a framework that offer many functionalities.

This a flip-side, the size of the gzipped file size is 143k, while for React is 43k and for Vue is 23K.

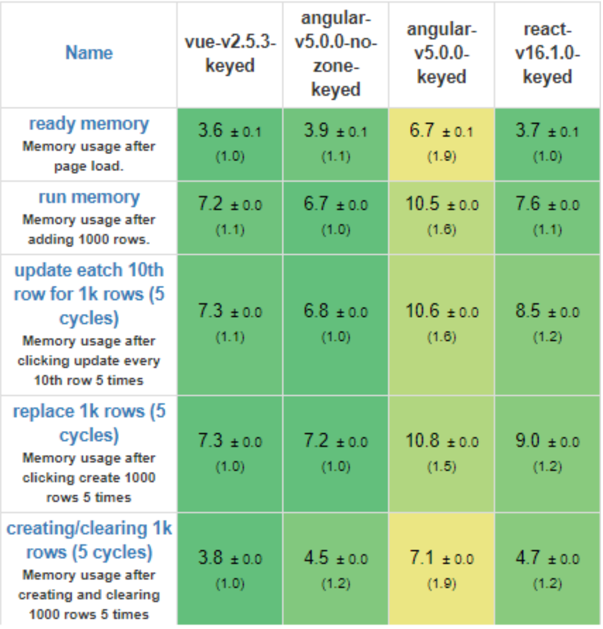
As regarding performance React and Vue use Virtual DOM that increase performance.

To compare the performance about speed and memory allocation we us the **js-framework-benchmark**, (available on gitHub) that compare many js library.



Name	vue-v2.5.3-keyed	angular-v5.0.0-no-zone-keyed	angular-v5.0.0-keyed	react-v16.1.0-keyed
create rows Duration for creating 1000 rows after the page loaded.	169.2 ± 3.6 (1.0)	170.9 ± 6.4 (1.0)	185.7 ± 7.8 (1.1)	201.2 ± 12.1 (1.2)
replace all rows Duration for updating all 1000 rows of the table (with 5 warmup iterations).	161.8 ± 3.9 (1.0)	176.6 ± 5.4 (1.1)	179.3 ± 6.5 (1.1)	169.0 ± 4.3 (1.0)
partial update Time to update the text of every 10th row (with 5 warmup iterations) for a table with 10k rows.	168.1 ± 7.4 (2.3)	73.7 ± 4.1 (1.0)	73.5 ± 4.9 (1.0)	90.9 ± 3.3 (1.2)
select row Duration to highlight a row in response to a click on the row. (with 5 warmup iterations).	9.8 ± 2.5 (1.0)	8.8 ± 3.5 (1.0)	7.6 ± 4.0 (1.0)	12.4 ± 4.1 (1.0)
swap rows Time to swap 2 rows on a 1K table. (with 5 warmup iterations).	19.0 ± 2.8 (1.0)	117.9 ± 2.7 (6.2)	118.5 ± 2.8 (6.2)	121.8 ± 4.2 (6.4)
remove row Duration to remove a row. (with 5 warmup iterations).	52.5 ± 1.8 (1.2)	43.5 ± 2.3 (1.0)	46.1 ± 2.6 (1.1)	51.5 ± 2.0 (1.2)
create many rows Duration to create 10,000 rows	1,521.4 ± 55.7 (1.0)	1,629.6 ± 53.4 (1.1)	1,682.0 ± 53.1 (1.1)	2,033.7 ± 32.0 (1.3)
append rows to large table Duration for adding 1000 rows on a table of 10,000 rows.	338.4 ± 10.3 (1.3)	275.4 ± 5.1 (1.1)	257.6 ± 11.1 (1.0)	271.8 ± 9.9 (1.1)
clear rows Duration to clear the table filled with 10.000 rows.	240.9 ± 11.4 (1.1)	334.7 ± 25.1 (1.5)	360.3 ± 16.4 (1.6)	224.4 ± 6.0 (1.0)
slowdown geometric mean	1.16	1.31	1.34	1.37

1 - Duration in milliseconds ± standard deviation (Slowdown = Duration / Fastest)



Name	vue-v2.5.3-keyed	angular-v5.0.0-no-zone-keyed	angular-v5.0.0-keyed	react-v16.1.0-keyed
ready memory Memory usage after page load.	3.6 ± 0.1 (1.0)	3.9 ± 0.1 (1.1)	6.7 ± 0.1 (1.9)	3.7 ± 0.1 (1.0)
run memory Memory usage after adding 1000 rows.	7.2 ± 0.0 (1.1)	6.7 ± 0.0 (1.0)	10.5 ± 0.0 (1.6)	7.6 ± 0.0 (1.1)
update eatch 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	7.3 ± 0.0 (1.1)	6.8 ± 0.0 (1.0)	10.6 ± 0.0 (1.6)	8.5 ± 0.0 (1.2)
replace 1k rows (5 cycles) Memory usage after clicking create 1000 rows 5 times	7.3 ± 0.0 (1.0)	7.2 ± 0.0 (1.0)	10.8 ± 0.0 (1.5)	9.0 ± 0.0 (1.2)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	3.8 ± 0.0 (1.0)	4.5 ± 0.0 (1.2)	7.1 ± 0.0 (1.9)	4.7 ± 0.0 (1.2)

2 - Memory allocation in MBs ± standard deviation

As you can see Vue generally outperforms many existing popular frameworks such as AngularJS or React, but all these frameworks are really pretty close to each other when compared to particularly slow or fast frameworks.

Trends

Another important aspect for comparing these three library is community support and popularity.

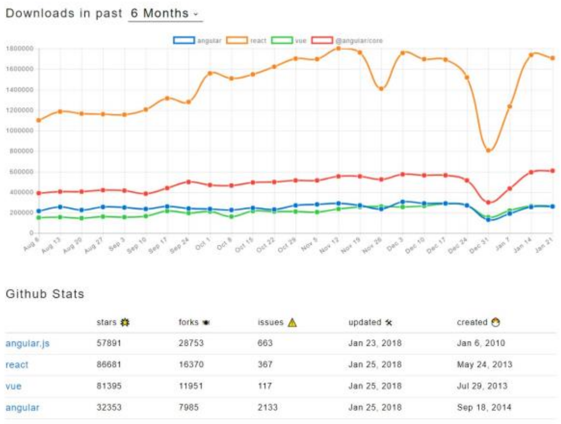
As mentioned before Angular and React have Google and Facebook behind.

This is definitely a big advantage for this two library compared to Vue.

Vue is less popular when compared to Angular and React, both of which have an impressive number of users and since there are fewer users of the framework, they share fewer solutions to extend the framework functionality (Fewer libraries for Vue.js).

Vue quickly became extremely popular in his native country China. As a result, documentation for some of the third-party libraries may be available only in Chinese and not many developers speak/read Chinese.

In the image below you can see the amount of downloads for the last 6 months (until 25 January 2018) and the GitHub statistics

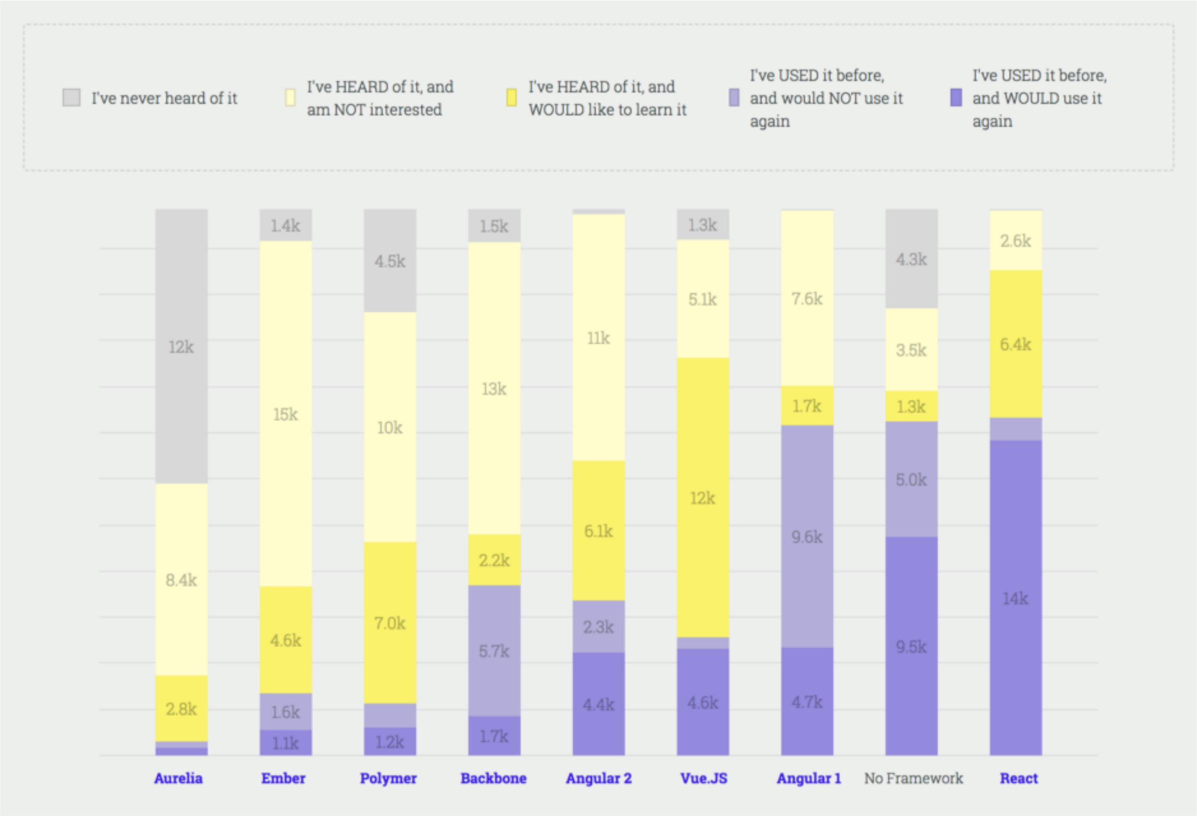


Github Stats

	stars	forks	issues	updated	created
angular.js	57891	28753	663	Jan 23, 2018	Jan 6, 2010
react	86681	16370	367	Jan 25, 2018	May 24, 2013
vue	81395	11951	117	Jan 25, 2018	Jul 29, 2013
angular	32353	7985	2133	Jan 25, 2018	Sep 18, 2014

It's interesting to see what developers think about this three technology.

In the 2017 edition of the annual State of JavaScript survey, collected from over 23,000 developers, we can see the actual state of satisfaction.



React is currently the dominant front-end library, while Vue it is the one that attracts the most attention.

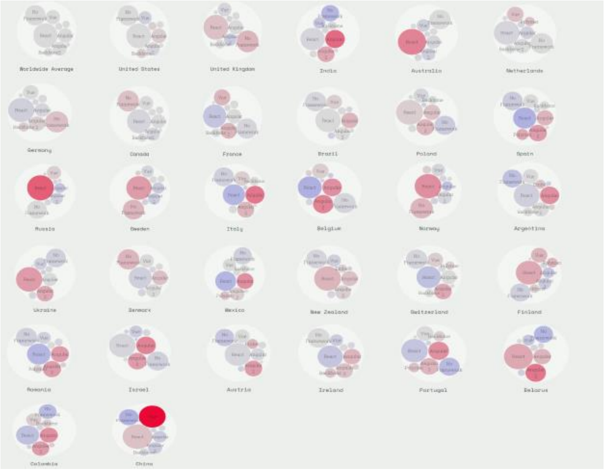
Angular don't have the same popularity as React but has Google and a huge user base behind.

But the final point is probably the most key: Angular is not trying to fight React head-to-head anymore, instead refocusing itself on the enterprise market

This result it's not true in all the world.

As you can see, it's true that there are major trends that hold true across regions (for example Angular React and Vue) but it’s also interesting to see that in each country the use of this technology is very different.

For example Vue is very popular in China, while Angular is most used in india.



Conclusion

It's really difficult to choose between this three library.

All three library have their pros and cons and none of them stands clearly above the others.

If you must build a solution with extensive and complex structure with a team that can be large, like an enterprise solution, Angular will be your choice.

Angular has a big community, which has been forming since the release of AngularJS and became even stronger when version 2 was released. This popularity means the availability of various solutions that are compatible with different versions of Angular, as well as the possibility of receiving advice from experienced developers and users without needing to contact the official support.

The downside is that , especially for a pure JavaScript developers, the learning curve is higher compared to other framework.

If you like Flexibility, choose between many library and be supported by the world major social media companies, than React is your choice.

JSX syntax and React philosophy can be difficult to understand for a developers (like me) when, for many years, the JavaScript logic and the UI Templates have been separated.

However it does not represent a big difficulty and with a bit of practice you get accustomed rather quickly to both the syntax and the type of programming.

React is very simple to learn, compared to Angular, and offers an impressive amount of library.

However, with great flexibility comes great responsibility.

There are no rules and limited guidance with React. Every project requires a decision regarding its architecture, and things can go wrong more easily.

Vue is one of the most rapidly growing JS frameworks, it's really performing and have a clean code.

His biggest problem, for me, it's the lack of a great community or company behind him.

Other developers say that Vue's small team is a benefit because it leads to cleaner and less over-engineered code but in a scenario like the development of enterprise applications this is a major risk.

Anyway Vue have a better story when it comes to offering a full framework-like experience (not as angular but better than react), thanks to official routing and state management libraries maintained by the same core team.

So this is it for now.

I actually find these new JavaScript Universe, concepts and opinions very interesting and fun.

I hope that each of you can analyze them and draw your own conclusions.

Thanks for reading!