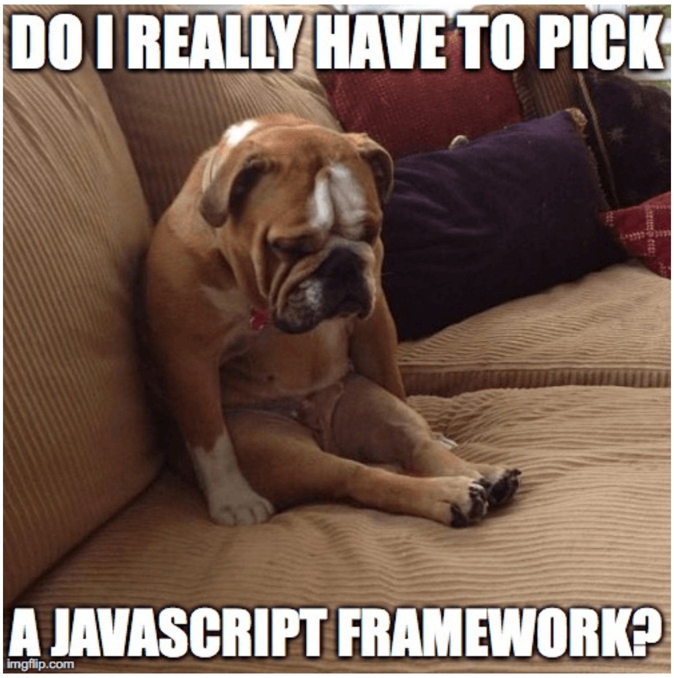
Table of Contents

- 1. Frameworks
    - 1.1. Vue.js
        - 1.1.1. Notizen
    - 1.2. Angular2
        - 1.2.1. Notizen
    - 1.3. aureliajs
- 2. Libraries
    - 2.1. React
        - 2.1.1. Pros
        - 2.1.2. Notizen
    - 2.2. jQuery
    - 2.3. Charts
    - 2.4. Entscheidungsmatrix
- 3. conclusion
- 4. TLDR:
    - 4.1. Vue.js
    - 4.2. warum nicht…
        - 4.2.1. Angular
        - 4.2.2. React
        - 4.2.3. jQuery
        - 4.2.4. aurelia

1 Frameworks

https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176 https://www.youtube.com/watch?v=z6hQqgvGI4Y https://about.gitlab.com/2016/10/20/why-we-chose-vue/ https://about.gitlab.com/2017/11/09/gitlab-vue-one-year-later/ https://about.gitlab.com/2017/02/06/vue-big-plan/ http://pixeljets.com/blog/why-we-chose-vuejs-over-react/ http://hire.jonasgalvez.com.br/2017/Jun/23/Why-Choose-Vue



*SO survey: https://insights.stackoverflow.com/survey/2017#technology-most-loved-dreaded-and-wanted-frameworks-libraries-and-other-technologies

1.1 Vue.js

https://vuejs.org/ https://github.com/vuejs/awesome-vue The Progressive JavaScript Framework

- HTML, CSS and JavaScript? start building things in no time!
- Versatile: scales between a library and a full-featured framework.
- Performant:
    - 20KB min+gzip Runtime
    - Blazing Fast Virtual DOM
    - Minimal Optimization Efforts
- shadow dom

1.1.1 Notizen

Vue is pretty easy to learn. Companies switch to Vue because its seems to be much easier for junior developers

Meanwhile, Angular and React have their own way of doing things. They may get in your way, because you need to adjust your practices to make things work their way. That can be a detriment because you are less flexible, and there is a steeper learning curve. It could also be a benefit because you are forced to learn the right concepts while learning the technology. With Vue, you can do the things the old-Javascript-fashioned way. This can be easier in the beginning, but could become a problem in the long-run if things are not done properly.

When it comes to debugging, it's a plus that React and Vue have less magic. The hunt for bugs is easier because there are fewer places to look and the stack traces have better distinctions between their own code and that of the libraries.

1.2 Angular2

https://angular.io/ One framework. Mobile & desktop.

- web, mobile web, native mobile and **native desktop**
- SPEED & PERFORMANCE

1.2.1 Notizen

There is definitely a steep learning curve for Angular.

loved by 52% of surveyed developers

The good thing about the Angular framework is that a new Angular 2 developer from another company will quickly familiarize themselves with all the requisite conventions. React projects are each different in terms of architectural decisions, and developers need to get familiar with the particular project setup.

1.3 aureliajs

http://aurelia.io/ Aurelia is a JavaScript client framework for web, mobile and desktop that leverages simple conventions to empower your creativity.

aurelia/framework Watch 549 Star 10,329 Fork 599

2 Libraries

with great flexibility comes great responsibility

2.1 React

https://reactjs.org/ A JavaScript library for building user interfaces

2.1.1 Pros

- shadow dom
- integrationen mit hooks
- React Native
- Component-Based (encapsulated)

2.1.2 Notizen

loved by 67% of surveyed developers

user "pier25" notes on Reddit that React makes sense if you are working for Facebook, where everyone is a superhero developer.

things they have done in React would have been better written in Vue. If you are an unexperienced Javascript developer — or if you worked mainly with jQuery in the last decade

2.2 jQuery

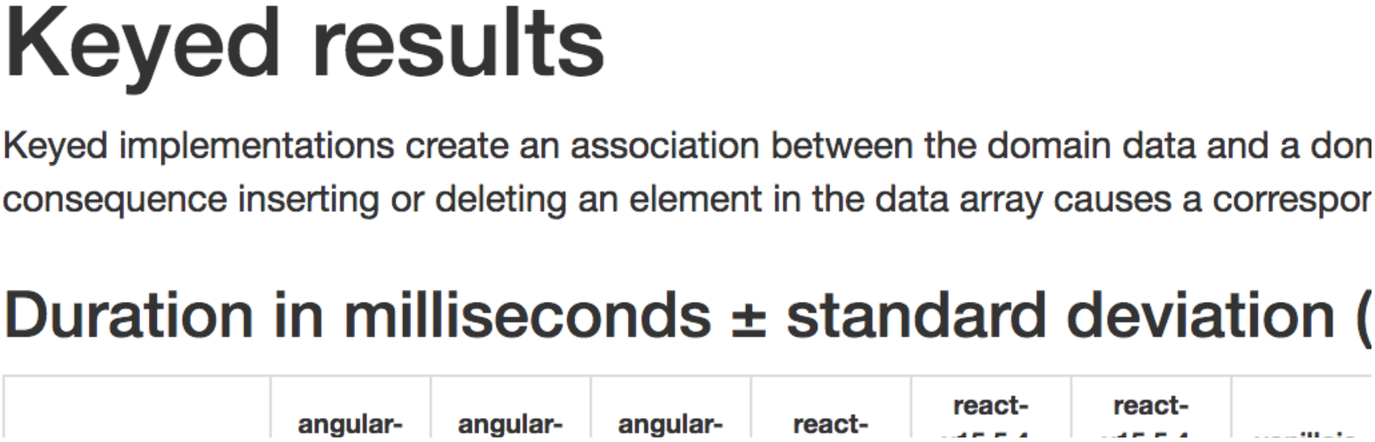
http://jquery.com/ dom & ajax

- Lightweight Footprint 32kB
- CSS3 Compliant
- Cross-Browser

2.3 Charts

Aug 28, 2017

#Fakenews: https://medium.com/@localvoid/how-to-win-in-web-framework-benchmarks-8bc31af76ce7



Keyed results

Keyed implementations create an association between the domain data and a don
consequence inserting or deleting an element in the data array causes a correspor

Duration in milliseconds ± standard deviation (

	angular-	angular-	angular-	react-	react-	react-

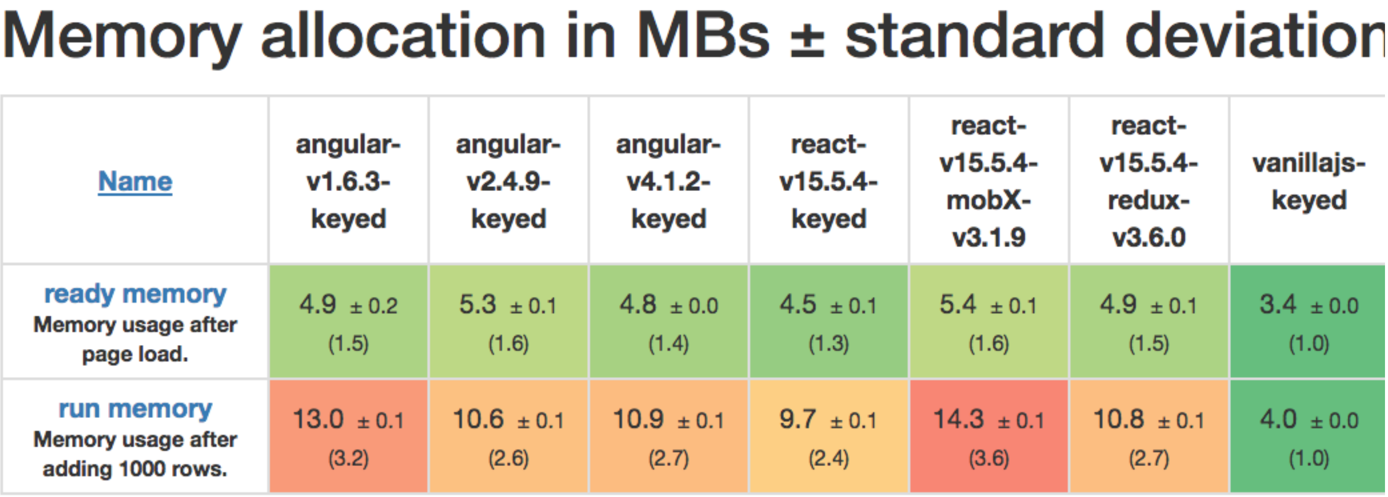
Name	v1.6.3-keyed	v2.4.9-keyed	v4.1.2-keyed	v15.5.4-keyed	v15.5.4-mobX-v3.1.9	v15.5.4-redux-v3.6.0	vanillajs-keyed
create rows Duration for creating 1000 rows after the page loaded.	251.8 ± 8.0 (1.8)	197.3 ± 9.3 (1.4)	193.1 ± 7.9 (1.4)	188.9 ± 10.9 (1.4)	243.9 ± 9.4 (1.8)	212.2 ± 14.2 (1.5)	138.5 ± 5.8 (1.0)
replace all rows Duration for updating all 1000 rows of the table (with 5 warmup iterations).	278.3 ± 16.7 (1.9)	201.3 ± 5.9 (1.4)	197.4 ± 5.3 (1.3)	201.0 ± 6.4 (1.4)	229.2 ± 12.2 (1.5)	206.7 ± 7.3 (1.4)	148.0 ± 4.5 (1.0)
partial update Time to update the text of every 10th row (with 5 warmup iterations).	12.5 ± 2.0 (1.0)	12.8 ± 3.3 (1.0)	13.0 ± 4.5 (1.0)	16.5 ± 2.3 (1.0)	16.0 ± 1.8 (1.0)	18.0 ± 1.6 (1.1)	14.1 ± 4.7 (1.0)
select row Duration to highlight a row in response to a click on the row. (with 5 warmup iterations).	8.1 ± 3.6 (1.0)	4.9 ± 3.4 (1.0)	3.4 ± 2.3 (1.0)	8.8 ± 3.4 (1.0)	10.1 ± 3.8 (1.0)	8.7 ± 2.9 (1.0)	10.1 ± 4.7 (1.0)
swap rows Time to swap 2 rows on a 1K table. (with 5 warmup iterations).	14.7 ± 1.5 (1.0)	13.5 ± 1.1 (1.0)	13.4 ± 1.0 (1.0)	14.7 ± 0.9 (1.0)	18.0 ± 1.2 (1.1)	17.1 ± 1.3 (1.1)	11.4 ± 1.1 (1.0)
remove row Duration to remove a row. (with 5 warmup iterations).	47.4 ± 2.4 (1.1)	46.4 ± 2.0 (1.1)	46.1 ± 3.2 (1.1)	47.2 ± 3.2 (1.1)	53.7 ± 2.1 (1.3)	52.4 ± 1.7 (1.2)	42.8 ± 1.9 (1.0)
create many rows Duration to create 10,000 rows	3108.7 ± 2162.2 (2.3)	1866.7 ± 55.4 (1.4)	1946.0 ± 41.8 (1.5)	1852.4 ± 29.0 (1.4)	2217.3 ± 71.5 (1.7)	1931.7 ± 35.6 (1.5)	1331.1 ± 22.2 (1.0)
append rows to large table Duration for adding 1000 rows on a table of 10,000 rows.	454.8 ± 42.1 (1.5)	365.1 ± 52.3 (1.2)	324.6 ± 10.1 (1.1)	345.6 ± 10.4 (1.2)	459.8 ± 47.2 (1.6)	366.4 ± 10.9 (1.2)	295.3 ± 12. (1.0)
clear rows Duration to clear the table filled with 10.000 rows.	817.6 ± 37.2 (4.7)	389.9 ± 39.0 (2.2)	379.9 ± 11.3 (2.2)	398.4 ± 8.2 (2.3)	495.1 ± 28.8 (2.8)	410.9 ± 9.8 (2.4)	174.8 ± 4.2 (1.0)
startup time Time for loading, parsing and starting up	118.1 ± 5.1 (2.9)	104.9 ± 8.6 (2.6)	84.3 ± 2.6 (2.1)	70.0 ± 2.9 (1.7)	87.6 ± 4.3 (2.2)	93.8 ± 6.9 (2.3)	40.5 ± 9.5 (1.0)
slowdown geometric mean	**1.69**	**1.36**	**1.31**	**1.30**	**1.51**	**1.41**	**1.00**

Non keyed results

Non keyed implementations are allowed to reuse existing dom elements. In consec row and update the contents of all elements after the inserting or deletion index. Th

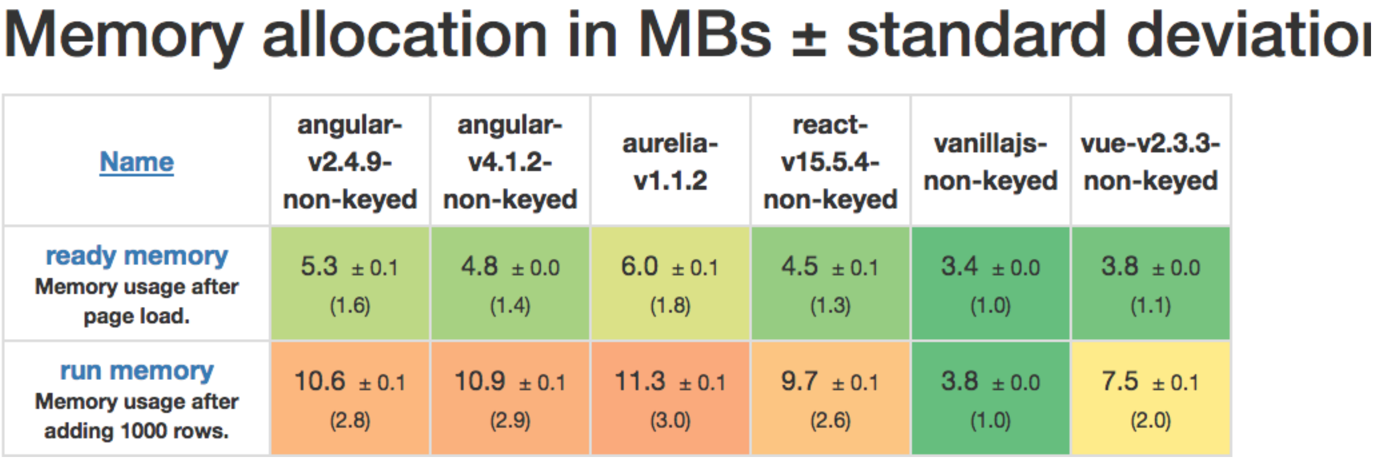
Duration in milliseconds ± standard deviation (

Name	angular-v2.4.9-non-keyed	angular-v4.1.2-non-keyed	aurelia-v1.1.2	react-v15.5.4-non-keyed	vanillajs-non-keyed	vue-v2.3.3-non-keyed
create rows Duration for creating 1000 rows after the page loaded.	198.3 ± 10.1 (1.4)	198.8 ± 10.1 (1.4)	185.4 ± 8.4 (1.3)	187.9 ± 9.0 (1.4)	137.8 ± 6.2 (1.0)	168.1 ± 10.1 (1.2)
replace all rows Duration for updating all 1000 rows of the table (with 5 warmup iterations).	57.8 ± 3.1 (1.0)	59.6 ± 4.8 (1.0)	89.8 ± 4.8 (1.6)	77.5 ± 2.3 (1.3)	64.0 ± 7.9 (1.1)	68.0 ± 3.4 (1.2)
partial update Time to update the text of every 10th row (with 5 warmup iterations).	12.4 ± 2.7 (1.0)	13.2 ± 3.8 (1.0)	12.2 ± 2.4 (1.0)	16.3 ± 2.6 (1.0)	12.4 ± 3.2 (1.0)	17.1 ± 1.5 (1.1)
select row Duration to highlight a row in response to a click on the row. (with 5 warmup iterations).	5.4 ± 3.8 (1.0)	4.1 ± 2.0 (1.0)	13.6 ± 1.6 (1.0)	8.2 ± 3.2 (1.0)	8.4 ± 4.3 (1.0)	10.5 ± 2.3 (1.0)
swap rows Time to swap 2 rows on a 1K table. (with 5 warmup iterations).	8.8 ± 0.9 (1.0)	8.7 ± 0.8 (1.0)	12.4 ± 1.2 (1.0)	11.6 ± 1.2 (1.0)	8.0 ± 0.8 (1.0)	14.5 ± 1.1 (1.0)
remove row Duration to remove a row. (with 5 warmup iterations).	34.0 ± 2.7 (1.0)	34.4 ± 2.6 (1.0)	59.7 ± 2.7 (1.8)	53.1 ± 2.4 (1.6)	35.8 ± 3.6 (1.1)	42.7 ± 2.0 (1.3)
create many rows Duration to create 10,000 rows	1875.2 ± 63.9 (1.4)	1974.4 ± 37.4 (1.5)	1875.2 ± 60.7 (1.4)	1851.4 ± 34.1 (1.4)	1346.4 ± 30.1 (1.0)	1587.8 ± 31.6 (1.2)
append rows to large table Duration for adding 1000 rows on a table of 10,000 rows.	361.6 ± 46.7 (1.2)	328.2 ± 11.3 (1.1)	328.9 ± 13.7 (1.1)	351.1 ± 13.5 (1.2)	294.1 ± 10.1 (1.0)	399.8 ± 10.1 (1.4)
clear rows Duration to clear the table filled with 10.000 rows.	392.4 ± 46.7 (2.2)	393.8 ± 9.9 (2.2)	285.5 ± 13.6 (1.6)	393.0 ± 9.5 (2.2)	175.3 ± 4.9 (1.0)	253.9 ± 5.6 (1.4)
startup time Time for loading, parsing and starting up	104.4 ± 2.3 (2.5)	83.3 ± 2.3 (2.0)	131.4 ± 3.1 (3.2)	68.3 ± 1.9 (1.7)	41.0 ± 8.7 (1.0)	58.2 ± 2.5 (1.4)
slowdown geometric mean	1.30	1.27	1.40	1.33	1.02	1.20



Memory allocation in MBs ± standard deviation

Name	angular-v1.6.3-keyed	angular-v2.4.9-keyed	angular-v4.1.2-keyed	react-v15.5.4-keyed	react-v15.5.4-mobX-v3.1.9	react-v15.5.4-redux-v3.6.0	vanillajs-keyed
ready memory Memory usage after page load.	4.9 ± 0.2 (1.5)	5.3 ± 0.1 (1.6)	4.8 ± 0.0 (1.4)	4.5 ± 0.1 (1.3)	5.4 ± 0.1 (1.6)	4.9 ± 0.1 (1.5)	3.4 ± 0.0 (1.0)
run memory Memory usage after adding 1000 rows.	13.0 ± 0.1 (3.2)	10.6 ± 0.1 (2.6)	10.9 ± 0.1 (2.7)	9.7 ± 0.1 (2.4)	14.3 ± 0.1 (3.6)	10.8 ± 0.1 (2.7)	4.0 ± 0.0 (1.0)



Memory allocation in MBs ± standard deviatio

Name	angular-v2.4.9-non-keyed	angular-v4.1.2-non-keyed	aurelia-v1.1.2	react-v15.5.4-non-keyed	vanillajs-non-keyed	vue-v2.3.3-non-keyed
ready memory Memory usage after page load.	5.3 ± 0.1 (1.6)	4.8 ± 0.0 (1.4)	6.0 ± 0.1 (1.8)	4.5 ± 0.1 (1.3)	3.4 ± 0.0 (1.0)	3.8 ± 0.0 (1.1)
run memory Memory usage after adding 1000 rows.	10.6 ± 0.1 (2.8)	10.9 ± 0.1 (2.9)	11.3 ± 0.1 (3.0)	9.7 ± 0.1 (2.6)	3.8 ± 0.0 (1.0)	7.5 ± 0.1 (2.0)

2.4 Entscheidungsmatrix

gh star history: http://www.timqian.com/star-history/#facebook/react&angular/angular&vuejs/vue

	Vue.js	React	Angular2	jQuery	vanilla.js
Watch	4,302	5,323	2,839	3,535	
Created	29.6.13	24.5.13	6.1.10	3.4.09	
Star	78,885	84,995	31,691	47,620	
Fork	11,666	16,092	7,867	14,690	14,069
contributors	120	1000	463	5	
Team	16	¯\(ツ)_/¯	36	9	
Vendor	Vue Technology LLC	Facebook	Google	jQuery Foundation	
Lizenz	MIT	MIT	MIT	MIT	
lang	js ES5 or ES6	js ES6, jsx	TypeScript	js	
loved by so*	¯\(ツ)_/¯	67 %	0,52 %	-	
dev Zeit	x				
Ökosystem		x			
Integrationen	x				
Zukunft	x	x	x	x	x
Performance	x	x	x	x	x
Schwerpunkt	x				

3 conclusion

What should I choose?

like really clean code	Vue

deadlines	Vue
want the easiest learning curve	Vue
want the most lightweight framework	Vue
want separation of concerns in one file	Vue
are working alone or have a small team	Vue (or React)
are working alone or have a small team	Vue (or React)
suspect fb & google	Vue
work at Google	Angular
love TypeScript	Angular (or React)
love object-orientated-programming (OOP)	Angular
need guidance, structure and a helping hand	Angular
work at Facebook	React
like flexibility	React
love big ecosystems	React
like choosing among dozens of packages	React
love JS & the “everything-is-Javascript-approach”	React
app tends to get really large	Angular (or React)
want to build an app with react-native	React
want to have a lot of developers in the pool	Angular or React
work with designers and need clean HTML files	Angular or Vue
like Vue but are afraid of the limited ecosystem	React

So, have you made your decision?

4 TLDR:

4.1 Vue.js

- gute Software
- Etwicklungszeit
- Lernkurve: HTML, CSS and JavaScript? start building things in no time!
- mächtig (reicht für Gitlab)
- flexibel: scales between a library and a full-featured framework.
- performant genug
- SPA
- zukunftssicher
- komponentenbasiert
- erfüllt die Checkliste

4.2 warum nicht…

4.2.1 Angular

- Angular1 = legacy
- Angular2 = TypeScript
- Lernkurve
- zu enterprise
- anängig von Microsoft & Google

4.2.2 React

- Lernkurve
- kein Framework
- JSX
- anhängig von Facebook

4.2.3 jQuery

- kein Framework
- Lernkurve

4.2.4 aurelia

- wie Vuejs nur nicht so beliebt

Author: Markus Bayer
Created: 2018-01-07 So 15:02
Validate