

**MITRAIS WHITE PAPER
COMPARING VueJS to
OTHER JAVASCRIPT WEB FRONT-END
FRAMEWORKS**

VER.1

Table of Contents

Table of Contents.....	2
1. JavaScript Framework.....	3
2. VueJS	3
3. AngularJS	4
4. ReactJS	6
5. EmberJS.....	8
6. Polymer.....	9
7. Benchmark	10
8. Keyed results	11
9. Non Keyed Results	12
10. Summary	13
Copyright	14

1. JavaScript Framework

Nowadays, the number of Javascript frameworks are growing rapidly. A few years ago, only a few JavaScript frameworks and libraries were easy to use when developing Web Applications. Often, we needed to hand-craft scripts to satisfy our requirements, and when we did, browser compatibility was a constant headache. Fortunately, most popular browsers have now adopted features designed to aid the execution of web applications. This has led to the birth of a huge range of front-end frameworks, and VueJS is one of the best.

2. VueJS

2.1. What is VueJS?

VueJS is a progressive JavaScript framework used to develop interactive web interfaces, with its focus mainly on the front end. It is very easy to integrate with other projects and libraries, so if integration is your priority, VueJS is a good choice.

2.2. Template

Vue recommends using templates to build HTML, and VueJS supports the full programmatic power of JavaScript. In VueJS, developers use a render function to generate a Virtual DOM into DOM. If the programmers have a strong knowledge of JavaScript, they can use the render function to build a Virtual DOM and parsing into DOM. But the template in VueJS will be generated as a Virtual DOM by VueJS. This makes VueJS one of fastest JS libraries to manipulate web pages. Like Angular, VueJS has built-in directives such as v-if, v-else, v-show, v-on, v-bind, and v-model.

VueJS uses HTML, JS and CSS separately, so if the developer already knows the basics of web development such as HTML, JS, and CSS, it is fast to learn.

2.3. Data Binding

The data binding features of VueJS help manipulate or assign values to HTML attributes, change the style, and assign classes with the help of the binding directive called v-bind.

2.4. Flexibility

The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. Because of this, VueJS is flexible. It is a view layer library that doesn't enforce any application-level architecture.

2.5. Components

VueJS supports component features. These features are very powerful and make re-use easy in different pages. When the programmer marks some block of the HTML tag is a component, they can easily reuse that component with no need to rebuild from the HTML tag.

2.6. Sample

```
<div id="app">
  {{ message }}
</div>
```

```
var app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue!'
  }
})
```

2.7. Preview

Hello Vue!

3. AngularJS

3.1. What is Angular?

AngularJS is an open source web application framework originally developed in 2009 by Misko Hevery and Adam Abrons. It is now maintained by Google.

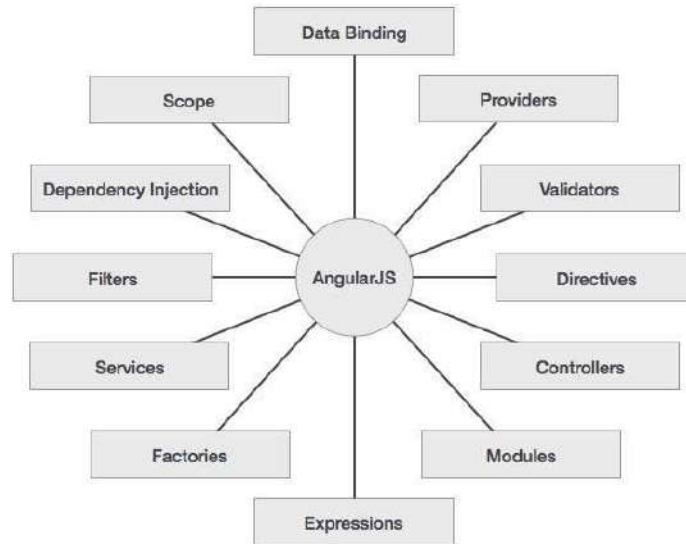
Angular is popular as it enforces a structured approach to development and comes with many useful libraries out of the box.

Angular relies on TypeScript, which was originally developed by Microsoft. Typescript is actually a syntactical superset of JavaScript, and includes static typing to assist developers. TypeScript is not standard JavaScript, and has a tiny user base compared to that of the whole JavaScript language.

AngularJS (Version 1.x) is a powerful JavaScript-based development framework to create RICH Internet Application (RIA), providing developers options to write client-side application (using JavaScript) in a clean MV*(Model View Whatever - MVC / MVVM / MVVC) way. Applications written in AngularJS are cross-browser compliant. AngularJS automatically handles JavaScript code suitable for each browser. But after the major upgrade to Version 2.x, Angular is moving its architecture from one based on MV* towards a component/services architecture. Following this upgrade, many applications developed in AngularJS needed to be rewritten to suit, which has persuaded many organisations to switch to other Javascript languages such as ReactJS.

AngularJS extends HTML with ng-directives. There are 3 attributes that can be used in HTML to define each function. The ng-app directive defines an AngularJS application. The ng-model directive binds the value of HTML controls (input, select, text area) to application data. The ng-bind directive binds application data to the HTML view.

3.2. Concept



3.3. Template

Angular templates are enhanced HTML with special Angular language and forces developers to learn Angular-specific syntax and to separate UI templates and inline JavaScript logic. On the template, views are pure HTML pages, and controllers written in JavaScript do the business processing. AngularJS only helps the developer to manipulate the original regular DOM into another regular DOM without changing it into virtual DOM. Care needs to be taken, since if there are many updates on the regular DOM, it will make for poor page display performance.

3.4. Data Binding

Angular is two-way-binding, changing the model state when the UI element (e.g. a user input) is updated. It uses automatic synchronisation of data between model and view components.

3.5. Pattern

Angular includes dependency injection, a pattern in which one object supplies the dependencies (a service) to another object (a client). This leads to flexibility and cleaner code.

3.6. Flexibility

Angular works best for SPA, and may be too "bloated" to be used for microservices. Features in Angular include data binding, scope, controller, services, filters, directives, templates, routing, etc.

3.7. Sample

```
<html>
  <head>
    <title>AngularJS First Application</title>
  </head>
  <body>
    <h1>Sample Application</h1>
  </body>
</html>
```

```
<div ng-app = "">
  <p>Enter your Name: <input type = "text" ng-model = "name"></p>
  <p>Hello <span ng-bind = "name"></span>!</p>
</div>

<script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

</body>
</html>
```

Preview

Sample Application

Enter your Name:

Hello Mitrais!

When a user types input text, after every keyboard release the text will be shown below the text box after the Hello label.

4. ReactJS

4.1. What is ReactJS?

ReactJS is JavaScript library used for building reusable UI components. Like VueJS, ReactJS is focused on the core library, leaving other features such as routing and global state management to companion libraries. The difference between ReactJS and other JavaScript libraries is that ReactJS puts HTML into JavaScript, while others put JavaScript into HTML. Programmers need to be strong in JavaScript if they want to use ReactJS as a frontend framework to build the UI.

4.2. Template

React uses JSX for templating instead of regular JavaScript. JSX looks like a regular HTML in most cases. ReactJS is faster because it performs optimisation while compiling code to JavaScript and type-safe, which traps many common errors during compilation.

4.3. Components

Like VueJS, ReactJS also supports component features. The developer defines the component via JavaScript, and can recall the component with other data on different pages. The rule is that component names in ReactJS should start with an uppercase character. It is easy for the developer to check manually whether the HTML tag is a ReactJS component or an HTML tag, since the HTML tag uses lowercase to define the tag name.

4.4. Native Rendering

React Native enables developers to write native-rendered apps for iOS and Android using the same React component model. This is great in that, as a developer, you can apply your knowledge of a framework across multiple platforms.

4.5. Sample

```
<!DOCTYPE html>
<html lang = "en">

  <head>
    <meta charset = "UTF-8">
    <title>React App</title>
  </head>

  <body>
    <div id = "app"></div>
    <script src = "index.js"></script>
  </body>

</html>
```

```
import React from 'react';

class App extends React.Component {
  render() {
    return (
      <div>
        Hello World!!!
      </div>
    );
  }
}
export default App;
```

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App.jsx';

ReactDOM.render(<App />, document.getElementById('app'));
```

4.6. Preview



5. EmberJS

5.1. What is EmberJS?

Ember.js is an open-source JavaScript client-side framework used for developing web applications. It uses the MVC (Model-View-Controller) architecture pattern. In Ember.js, the route is used as the model, handlebar template as the view, and the controller manipulates the data in the model.

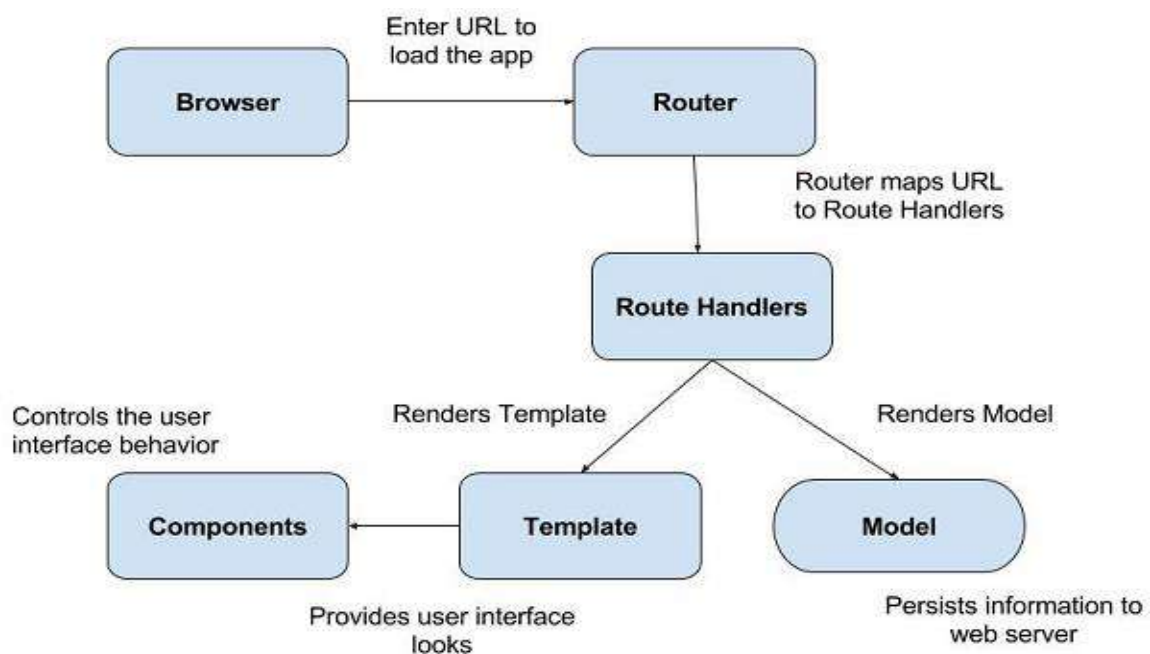
5.2. Components

EmberJS component uses the feature from the HandlebarsJS model. Just define part of HTML tag, identify them as a class or id, and create a handlebars template with both combined in the model.

5.3. Template

EmberJS use a template library called Handlebars. Handlebars is one of the powerful javascript template engines forking from MustacheJS. Handlebars can compile part or all of the HTML tag, and can be reused for separate pages.

5.4. Core Concept



5.5. Sample

app/templates/components/blog-post.hbs

HBS

```
1 <article class="blog-post">
2   <h1>{{title}}</h1>
3   <p>{{yield}}</p>
4   <p>Edit title: {{input type="text" value=title}}</p>
5 </article>
```

app/templates/index.hbs

HBS

```
1 {{#each model as |post|}}
2   {{#blog-post title=post.title}}
3     {{post.body}}
4   {{/blog-post}}
5 {{/each}}
```

app/routes/index.js

JS

```
1 import Route from '@ember/routing/route';
2
3 export default Route.extend({
4   model() {
5     return this.get('store').findAll('post');
6   }
7 });
```

6. Polymer

6.1. What is Polymer?

Polymer is a JavaScript library created by Google that allows reuse of HTML elements for building applications with components. Polymer 2.x use ECMAScript 2015 (commonly known as ES6). However, major browsers are mostly only compatible with ES5.

6.2. Components

The component model in Polymer looks like the component model in VueJS. In fact, it was a source of inspiration for Vue. The difference in Polymer is that it is built upon the latest Web Components features, so the developer can create reusable custom elements that interoperate seamlessly with the browser's built-in elements. Alternatively, the developer can break the app up into right-sized components, making code cleaner and less expensive to maintain.

6.3. Deploy To Production

Polymer develops the code use ES6. Because most browsers remain incompatible with ES6, code should compile into ES5 to make it as compatible as possible. Compiling the source into ES5, the compiler tools in Polymer can help developers to optimise and minify the code.

6.4. Sample

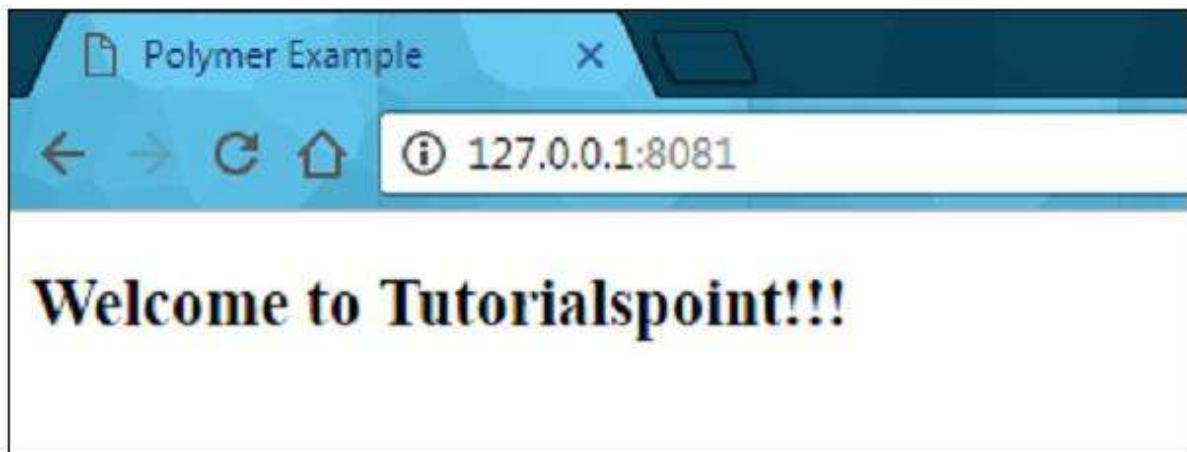
```
<!doctype html>
<html lang = "en">
  <head>
    <title>Polymer Example</title>
    <script src = "bower_components/webcomponentsjs/webcomponents-lite.js"></script>
    <link rel = "import" href = "bower_components/polymer/polymer.html">
    <link rel = "import" href = "define-element.html">
  </head>

  <body>
    <define-element></define-element>
  </body>
</html>
```

```
<dom-module id = "define-element">
  <template>
    <h2>Welcome to Tutorialspoint!!!</h2>
  </template>

  <script>
    HTMLImports.whenReady(function(){
      Polymer ({
        is: "define-element"
      })
    })
  </script>
</dom-module>
```

6.5. Preview



7. Benchmark

Whilst completing this JavaScript framework comparison, we have done some benchmarking. The results may make you think twice when choosing a framework. This benchmark was run on a MacBook Pro 15 (2.5 GHz i7, 16 GB RAM, OSX 10.13.1, Chrome 62.0.3202.94 (64-bit)). Using other browsers, results may vary.

8. Keyed results

Associations between domain data and DOM elements can be made by assigning a 'key'. This means that when data changes, the DOM element with a matching key will be updated. Any inserts or deletes of an element in the data array triggers a change to the DOM.

Duration in milliseconds ± standard deviation (Slowdown = Duration / Fastest)					
Name	angular-v5.2.2-optimized-keyed	angular-v5.2.2-no-zone-keyed	reactive-edge-keyed	ember-v3.0.0-keyed	react-lite-v0.15.30-keyed
create rowsDuration for creating 1000 rows after the page loaded.	159.35.3 (1.2)	164.66.6 (1.2)	299.213.7 (2.2)	399.119.9 (2.9)	169.07.8 (1.2)
replace all rowsDuration for updating all 1000 rows of the table (with 5 warmup iterations).	167.05.2 (1.1)	163.75.9 (1.1)	281.920.5 (1.9)	291.637.4 (2.0)	228.95.1 (1.5)
partial updateTime to update the text of every 10th row (with 5 warmup iterations) for a table with 10k rows.	71.82.1 (1.0)	70.81.8 (1.0)	81.63.7 (1.2)	151.116.1 (2.1)	2,019.549.3 (28.5)
select rowDuration to highlight a row in response to a click on the row. (with 5 warmup iterations).	7.23.7 (1.0)	10.04.4 (1.0)	11.42.4 (1.0)	9.32.6 (1.0)	23.00.7 (1.4)
swap rowsTime to swap 2 rows on a 1K table. (with 5 warmup iterations).	19.05.0 (1.2)	109.41.0 (6.8)	20.52.9 (1.3)	134.73.3 (8.4)	147.04.8 (9.2)
remove rowDuration to remove a row. (with 5 warmup iterations).	48.52.5 (1.1)	44.81.5 (1.0)	54.82.6 (1.2)	57.02.2 (1.3)	65.32.0 (1.5)
create many rowsDuration to create 10,000 rows	1,423.171.8 (1.1)	1,419.673.7 (1.1)	2,487.092.7 (1.9)	2,454.258.5 (1.9)	2,281.327.9 (1.7)
append rows to large tableDuration for adding 1000 rows on a table of 10,000 rows.	234.613.0 (1.1)	244.25.8 (1.1)	375.476.8 (1.7)	367.18.9 (1.7)	2,523.655.6 (11.7)
clear rowsDuration to clear the table filled with 10,000 rows.	276.715.2 (1.6)	292.72.9 (1.7)	550.578.0 (3.2)	266.56.3 (1.6)	308.19.4 (1.8)
slowdown geometric mean	1.14	1.39	1.63	2.05	3.23

Startup metrics					
Name	angular-v5.2.2-optimized-keyed	angular-v5.2.2-no-zone-keyed	reactive-edge-keyed	ember-v3.0.0-keyed	react-lite-v0.15.30-keyed
consistently interactive pessimistic TTI - when the CPU and network are both definitely very idle. (no more CPU tasks over 50ms)	62.00.7 (1.5)	61.10.7 (1.5)	114.81.6 (2.8)	200.13.5 (4.9)	85.14.0 (2.1)
script bootstrap timethe total ms required to parse/compile/evaluate all the page's scripts	14.30.4 (1.0)	13.80.2 (1.0)	36.52.0 (2.3)	74.32.0 (4.6)	10.50.5 (1.0)
main thread work costtotal amount of time spent doing work on the main thread. includes style/layout/etc.	166.70.9 (1.1)	166.01.3 (1.1)	179.14.0 (1.2)	221.12.3 (1.5)	153.72.9 (1.0)
total byte weightnetwork transfer cost (post-compression) of all the resources loaded into the page.	270,488,00.0 (1.7)	266,304,00.0 (1.6)	371,515,00.0 (2.3)	749,206,00.0 (4.6)	192,135,00.0 (1.2)

Memory allocation in MBs ± standard deviation					
Name	angular-v5.2.2-optimized-keyed	angular-v5.2.2-no-zone-keyed	reactive-edge-keyed	ember-v3.0.0-keyed	react-lite-v0.15.30-keyed
ready memoryMemory usage after page load.	3.50.2 (1.6)	3.50.1 (1.6)	4.50.1 (2.1)	7.40.2 (3.4)	3.40.1 (1.6)
run memoryMemory usage after adding 1000 rows.	6.20.0 (2.3)	6.30.0 (2.3)	18.90.1 (7.0)	16.50.0 (6.1)	13.40.0 (4.9)
update each 10th row for 1k rows (5 cycles)Memory usage after clicking update every 10th row 5 times	6.30.0 (2.2)	6.40.0 (2.2)	18.90.0 (6.4)	16.60.0 (5.7)	7.80.0 (2.6)
replace 1k rows (5 cycles)Memory usage after clicking create 1000 rows 5 times	6.70.0 (2.2)	6.80.0 (2.2)	19.30.0 (6.3)	17.50.0 (5.7)	7.90.0 (2.6)
creating/clearing 1k rows (5 cycles)Memory usage after creating and clearing 1000 rows 5 times	4.10.0 (1.3)	4.10.0 (1.5)	6.30.0 (2.3)	8.90.0 (3.2)	3.70.0 (1.4)

9. Non Keyed Results

Duration in milliseconds ± standard deviation (Slowdown = Duration / Fastest)							
Name	angular-v5.2.2-non-keyed	react-v16.1.0-non-keyed	vue-v2.5.3-non-keyed	vuera-v0.1.3-non-keyed	ractive-v0.9.9-non-keyed	ractive-edge-non-keyed	polymer-v2.0.0-non-keyed
create rowsDuration for creating 1000 rows after the page loaded.	178.07.8 (1.3)	187.44.2 (1.4)	175.74.8 (1.3)	166.18.3 (1.2)	300.715.4 (2.2)	291.410.7 (2.2)	203.17.0 (1.5)
replace all rowsDuration for updating all 1000 rows of the table (with 5 warmup iterations).	55.83.0 (1.0)	67.02.4 (1.2)	66.22.5 (1.2)	65.12.5 (1.2)	63.53.7 (1.1)	66.31.5 (1.2)	62.01.7 (1.1)
partial updateTime to update the text of every 10th row (with 5 warmup iterations) for a table with 10k rows.	69.72.8 (1.0)	91.95.9 (1.3)	160.57.6 (2.3)	170.214.2 (2.4)	79.74.1 (1.1)	82.13.6 (1.2)	408.717.7 (5.9)
select rowDuration to highlight a row in response to a click on the row. (with 5 warmup iterations).	11.02.9 (1.0)	10.14.5 (1.0)	9.93.5 (1.0)	10.21.6 (1.0)	9.54.1 (1.0)	9.64.1 (1.0)	10.74.6 (1.0)
swap rowsTime to swap 2 rows on a 1K table. (with 5 warmup iterations).	14.85.5 (1.0)	12.74.8 (1.0)	14.62.1 (1.0)	16.94.4 (1.1)	14.93.5 (1.0)	12.74.0 (1.0)	15.64.6 (1.0)
remove rowDuration to remove a row. (with 5 warmup iterations).	33.63.9 (1.1)	42.81.9 (1.4)	40.81.8 (1.4)	42.33.4 (1.4)	41.92.4 (1.4)	43.12.2 (1.5)	47.02.5 (1.6)
create many rowsDuration to create 10,000 rows	1,526.759.3 (1.1)	2,039.945.4 (1.5)	1,586.626.2 (1.2)	1,598.131.7 (1.2)	2,449.694.0 (1.8)	2,449.796.9 (1.8)	2,244.050.8 (1.7)
append rows to large tableDuration for adding 1000 rows on a table of 10,000 rows.	231.83.6 (1.1)	262.06.7 (1.2)	326.17.7 (1.5)	332.85.3 (1.5)	347.870.1 (1.6)	360.078.4 (1.6)	605.730.8 (2.7)
clear rowsDuration to clear the table filled with 10,000 rows.	306.63.1 (1.8)	230.27.6 (1.3)	247.54.4 (1.4)	242.53.7 (1.4)	536.919.3 (3.1)	536.525.9 (3.1)	230.57.1 (1.3)
slowdown geometric mean	1.14	1.25	1.32	1.33	1.49	1.51	1.66

Startup metrics							
Name	angular-v5.2.2-non-keyed	react-v16.1.0-non-keyed	vue-v2.5.3-non-keyed	vuera-v0.1.3-non-keyed	ractive-v0.9.9-non-keyed	ractive-edge-non-keyed	polymer-v2.0.0-non-keyed
consistently interactivea pessimistic TTI - when the CPU and network are both definitely very idle. (no more CPU tasks over 50ms)	98.51.1 (2.4)	97.12.7 (2.4)	90.71.4 (2.2)	126.92.8 (3.1)	114.83.4 (2.8)	115.63.0 (2.8)	119.22.3 (2.9)
script bootuptime the total ms required to parse/compile/evaluate all the page's scripts	45.60.6 (2.8)	18.60.5 (1.2)	14.80.7 (1.0)	45.61.5 (2.8)	36.51.6 (2.3)	36.51.6 (2.3)	29.31.0 (1.8)
main thread work costtotal amount of time spent doing work on the main thread. includes style/layout/etc.	206.43.3 (1.4)	163.24.7 (1.1)	158.22.0 (1.1)	187.92.6 (1.3)	179.84.7 (1.2)	178.43.2 (1.2)	174.52.9 (1.2)
total byte weightnetwork transfer cost (post-compression) of all the resources loaded into the page.	306,208.00.0 (1.9)	263,080.00.0 (1.6)	221,603.00.0 (1.4)	454,731.00.0 (2.8)	370,850.00.0 (2.3)	371,374.00.0 (2.3)	344,552.00.0 (2.1)

Memory allocation in MBs ± standard deviation							
Name	angular-v5.2.2-non-keyed	react-v16.1.0-non-keyed	vue-v2.5.3-non-keyed	vuera-v0.1.3-non-keyed	ractive-v0.9.9-non-keyed	ractive-edge-non-keyed	polymer-v2.0.0-non-keyed
ready memoryMemory usage after page load.	6.20.0 (2.9)	3.70.1 (1.7)	3.50.1 (1.6)	5.00.1 (2.3)	4.50.1 (2.1)	4.50.1 (2.1)	3.80.1 (1.8)
run memoryMemory usage after adding 1000 rows.	10.10.0 (3.7)	7.60.0 (2.8)	7.00.0 (2.6)	8.60.0 (3.2)	18.90.1 (7.0)	18.90.0 (7.0)	6.50.2 (2.4)
update each 10th row for 1k rows (5 cycles)Memory usage after clicking update every 10th row 5 times	10.10.0 (3.5)	8.50.0 (2.9)	7.10.0 (2.4)	8.70.0 (2.9)	18.90.1 (6.4)	18.90.1 (6.4)	6.70.0 (2.3)
replace 1k rows (5 cycles)Memory usage after clicking create 1000 rows 5 times	10.30.0 (3.1)	11.80.0 (3.6)	7.10.0 (2.2)	8.70.0 (2.6)	19.00.0 (5.8)	19.00.0 (5.8)	6.80.1 (2.1)
creating/clearing 1k rows (5 cycles)Memory usage after creating and clearing 1000 rows 5 times	6.60.0 (2.4)	4.70.0 (1.7)	3.70.0 (1.3)	5.20.0 (1.9)	6.30.0 (2.3)	6.30.0 (2.3)	4.20.0 (1.5)

10. Summary

So here is our summary of the advantages and disadvantages.

- If the developer is experienced in JavaScript and your application is small enough, ReactJS is a great choice.
- If the primary concern is to take advantage of the newest browser features, Polymer might right.
- If we are building a big application with a complete feature set and based on an architecture like MVC, Angular looks like the best choice.
- If a developer only has basic exposure to web development, VueJS or EmberJS may be a better fit.
- Looking at the benchmarks above VueJS clearly comes out on top for performance, startup time and memory allocation

Copyright

Copyright © 2018 Mitrais

All rights reserved.

Disclaimer

Any and all information in this document has been compiled and provided for information purposes only. The information provided herein may include information compiled from a variety of third parties. Mitrais will not be liable for any loss, damage, cost or expense incurred in relation to or arising by reason of any person relying on the information in this document or any link to any website provided herein, whether or not caused by negligence on Mitrais' part. While Mitrais endeavours to provide the information up to date and correct, Mitrais make no representations or warranties of any kind, express or implied, about the accuracy, reliability, completeness or currency of the information or its usefulness in achieving any purpose. Readers of this document are responsible for assessing its relevance and verifying the accuracy of the content. In any event, the information provided herein should not be construed as providing advice whether legal or otherwise.