

Notes on "Your Best Options to Build Modern Front Ends with JavaScript" by Dan Wilson.

[Notes on "Your Best Options to Build Modern Front Ends with JavaScript" by Dan Wilson.](#)

Posted by [Andy Peterson](#)

13 Jul 2020 12:37 PM

Dan Wilson is fun to listen to and this was a nice presentation to acquaint the user with the differences between Angular, React, and Vue. He's been a member of the CF Community for years. I am not certain he uses it much nowadays.

Nevertheless, he is always informative, and if you are looking to adopt something other than jQuery to go along with your modernization initiative and are leaning towards another option, Dan does a great job of reviewing them.

https://www.youtube.com/watch?v=2_jWk-f9cI

Why you should watch

The three top vibrant frameworks that will still be around in five years are Angular, React, and Vue.js.

The problem is, many people still use jQuery. Is jQuery really that bad? :-)

What about Angular? Angular 1 is a totally different animal compared to later versions.

Angular was Angular.js. In CF Terms - it is kinda like ColdBox in the sense that it will do everything you need and there is "one true way." However, I do know a lot of Ortus folks that use Vue.js. Angular works for Google and there are other contributors as well.

Electron is for the desktop, and we need to look at mobile as well.

Doing CRUD with Angular took Dan three hours. Did last of the three. Found to be the most uncomfortable framework. Still left uncomfortable.

He created a Tour of Heroes site, and who hasn't?

This is the second time looking at angular code, while there is unfamiliar syntax, there is a lot done in a little amount of code.

There is a lot of importing going on, and the importing gives you exactly what you need.

If I understand correctly, what you're writing is typescript, in some cases, which is a transpiler. TypeScript is extremely popular and is a clean code creator as stated by those who use it.

What about React.js?

Well, it's a lot like custom tags, which is encapsulated UI. A lot of parallels there. Not a framework according to them, but a library. A big focus on UI, while the community supports the rest. ReactNative, NativeScript, and other options for Mobile and Desktop (Desktop Support: React Native UWP, MacOS, Electron)

So he built a tic tac toe game with React - it was a tutorial that took him about 2 hours. An internal state processing component reverted state. Again, you are importing your dependencies. Also, it uses "JSX" and uses class names because while it looks like html, it is JS. React is controlled by FB.

What about Vue.js?

I'm a framework, and grew really fast. Let's call it FW/1 for those CF devs out there. More open source (not controlled by Google or FB). Super vibrant community.

The tutorial took an hour mainly due to a progressive process of skill. A components model. Basically writing html and then decorating it vs a full on application may create . architectural issues later. Like jQuery, so maybe I'd like it.

Vue.js sits right in the middle of what is out of the box and what you can choose. Commitment to a single vendor? That'd be angular. So how do you view the future? React has the best app quality across platforms. Not really a bad choice.

First hour experience: React, then Vue, then Angular.

Enforce consistency thru app? Winner: Angular, then Vue, then React

But what about the "State of JavaScript" charting? Looks like React is basically stable and like it and are using it. More are using angular and liking it less. Vue is liked almost as much as react and used almost as much too.

Overall "winner" appears to be react, then vue, then angular.

Comments: In the angular camp, a lot of people like the separation, and lots of teams working on the codebase with an accepted way to do things.

Only framework that looked like html was Vue.

JSX was wierd to Dan but worth learning. Dan started out with System 1 and System 2 learning from a book called Thinking, Fast & Slow by Daniel Kahneman.

Jeffrey Houser has a book on angular with cfml back end which would be helpful to cfml devs.

And so what about Node.js? Definitely cool to run with JS front to back end. I was wondering about where Node.js fit in as well.

Tags:

[0 Comments](#)

Copyright © Andy Peterson. All rights reserved.

Powered by ContentBox v4.2.1+737