

Comparison of Frontend JavaScript Frameworks

React vs. Angular vs. Vue

Welcome

- * Thanks so much for being here and thanks for letting me come talk to you all about frontend frameworks
- * This is a topic that's near and dear to my heart
- * I started with learning HTML and then JavaScript, so I always like to find excuses to play around with React, Angular, and Vue

Introduction

Tell me about yourself

- * Who is already using one of these frameworks?
- * Who is completely happy with their choice?
- * Who is looking to make a change?
- * Who has been using React for over a year? Angular? Vue?
 - * I need your help!
 - * If I misspeak or do anything while coding that doesn't make any sense or you would do differently please call me out on it!

Why are you here?

- * Are you looking to learn your first framework?
- * Are you looking to improve your marketability?
- * Are you trying to solve a problem in your current situation?

Goals

- * The goal of this talk isn't to tell you everything that you need to know about these frameworks in order to be successful
 - * It wouldn't be possible to do that for just one of these frameworks let alone three
- * The goal is to give you enough familiarity with each of the frameworks so that you can decide which you may want to take a closer look at
 - * Or if you're already using a framework, whether or not it'd make sense for you to take a closer look at a different framework
- * My personal goal is not make too much of a fool of myself as I flail from one framework to another

Out of Scope

There's so much to talk about... something's got to go!

Routing
State Management
Universal Rendering
Mobile Development

We're not going to have time to talk any of these things

Framework vs Library

- * Angular and Vue are frameworks and React is a library
- * For simplicity sake, when I refer to React as a framework I'm referring to its larger ecosystem

Shared Concepts

Concepts

What are the conceptual things that you need to know in order to be successful with any of these frameworks?

SPAs

* Will you be building a single-page application?

* Or just adding some additional interactivity to server-rendered pages?

Components

- * Effectively leveraging components
- * Knowing how to break an application into components

Templating and Binding

- * Where and how are templates defined?
- * Binding syntax
- * Rendering values
- * Rendering lists
- * Handling events

Virtual DOM

- * React and Vue both utilize a Virtual DOM
- * In-memory representation of the physical DOM
- * When the state of your app changes, comparisons are done to the virtual DOM in order to determine how the physical DOM should be updated

Data Flow

- * How does data flow between components?
- * To child components
- * To parent components

State Management

- * Where does state live your application?
- * How is that state managed?
- * Redux is a popular way of managing state
 - * Supported in all three frameworks
 - * Not the only way

Backend APIs

- * AJAX
- * HTTP semantics
- * Authentication (OpenID Connect, OAuth, JWTs)

Routing

- * How are routes defined?
- * How do you redirect the user to another route?
- * Are sub or nested routes supported?

Language Abstractions

- * HTML (JSX)
- * CSS (Sass)
- * JavaScript (TypeScript)

Testing

- * Unit testing
- * E2E testing

Front End Builds

- * webpack is the defacto tool these days (sorry Grunt and Gulp)
- * Linting
- * Tree Shaking
- * Bundling
- * CLIs

(Simple) App Demos

Focus on the following...

- * Getting started
 - * Project creation
 - * Project layout
 - * Dev workflow
- * Basic components
 - * Templating
 - * Properties
 - * Events

Highlights

React

Pros - Relatively simple, elegant in its design, focus on functional programming, broad community support

Cons - Can feel a bit constraining or foreign at times, many ways to accomplish the same thing (i.e. less opinionated), you can't leverage native HTML (and sometimes even CSS)

Angular

Pros - Capable CLI, all inclusive, opinionated,
leverages native HTML and CSS

Cons - Relatively verbose, can feel heavy handed or
overly complex at times, difficult to get started without
the assistance of the CLI

Vue

Pros - Downgraded mode is brilliant, less opinionated
(especially when compared with Angular),
leverages native HTML and CSS

Cons - Less community support (at this time),
no "one way" to setup projects

Other Thoughts

Workflow

- * Think about your development, testing, and production workflows
- * Which framework will work best for your situation?
- * Will one of the CLIs work for your use case?
- * Or will you need to handle your frontend builds manually as part of your CI/CD process?

CLIs

- * Compare the three CLIs
- * Strengths? Weaknesses?
- * Angular CLI
 - * Initial project setup
 - * Components and services
 - * Dev server
 - * Build for production
- * Create React App
 - * Initial project setup
 - * Dev server
 - * Build for production
- * Vue CLI
 - * Initial project setup
 - * Dev server
 - * Build for production

Forms

- * Most devs tend to build lots of forms
- * Angular offers the most built-in for forms and form validation
- * Each framework also has libraries for building forms

Performance

Is one framework the clear leader in performance?

<http://www.stefankrause.net/js-frameworks-benchmark7/table.html>

Universal or Server-side Rendering

Mobile Dev

- * React Native
- * NativeScript

Learning

- * Available resources

- * Treehouse

- * Egghead

- * Pluralsight

- * Official docs

- * Quality of the docs counts

- * Vue is known for its good docs

- * But honestly I feel like all three are good

Tooling Support

- * Editors
 - * VS Code
 - * Atom
 - * Visual Studio
- * Debuggers (In-Browser)
- * Control Vendors
- * Online Tools
 - * <https://codesandbox.io/> <-- this is pretty cool
 - * <https://stackblitz.com/>

TypeScript Support

- * First class support in Angular
 - * The official docs use TS
- * React and Vue both offer TS support but it might be more difficult to get started and support ongoing

Popularity

Who's using these frameworks?

- * Angular
 - * Google
 - * <https://www.madewithangular.com/>
- * React
 - * Facebook
 - * Instagram
 - * Twitter
 - * Airbnb
 - * Netflix
- * Vue
 - * Expedia
 - * Nintendo
 - * IBM
 - * GitLab
 - * Sainsbury's

Support

They're all open source projects... so GitHub issues :)

Community

- * Local meetups
- * Conferences
- * Online communities

Wrap Up

How do you decide on which to use?

- * What does your dev team look like?
- * What kind of app are you building?
- * What features are most important to you?

My Experience

- * Using the downgraded Vue experience... is just fun
- * I feel like React encouraged me to break things down into smaller and smaller components
- * Angular feels the most C# like... thanks to TypeScript

Your Experience

What has your experience been like?

Advice

- * Are you new to frontend frameworks?
 - * Probably start with Vue
 - * Get comfortable with the core concepts
 - * Then go from there
- * Are you experienced?
 - * Review your requirements and team experience
 - * Determine which would be the best fit from there

Conclusion

- * You can be successful with any of these frameworks
- * Things will continue to change and evolve
- * Picking the "right" framework is important, but it's more important to focus on the core concepts and skills that front end devs need to have in order to be successful
 - * That will also equip you for whatever comes next... because you better believe that something new is right around the corner
- * Don't worry too much about it... just make a choice and do your best :)

Thanks!

James Churchill

Treehouse: <https://teamtreehouse.com/jameschurchill>

Twitter: @SmashDev

GitHub: smashdevcode