Front-end Frameworks

Traditional Web Apps and Single Page Apps (SPAs)

Choose Between Traditional Web Apps and Single Page Apps (SPAs)

Comparison of JavaScript Frameworks

Questions

1. How mature is it and who’s behind it?
2. What kind of features does it have?
3. What architecture, development paradigms, and patterns does it employ?
4. What is the ecosystem around it?
5. Will I and my colleagues be able to learn this tool with ease? **What does the learning curve look like for each framework?**
6. Does is fit well with my project?
7. What kind of performance can you expect from the frameworks?
8. Are the frameworks likely to be around for a while?
9. How extensive and helpful are their corresponding communities?

Resources

1. Front-end Frameworks - StatofJs 2018
2. The Ultimate Guide to JavaScript Frameworks
3. VueJS vs Angular: Angular (Formerly known as Angular 2)
4. StackOverflow - Developer Survey ResultsFrameworks, Libraries, and Tools 2018
5. React vs. Angular vs. Vue.js: A Complete Comparison Guide - DZone
6. React vs Angular: An In-depth Comparison - SitePoint

NPM Downloads

- angular vs react vs vue vs @angular/core

Conclusion

Advantages

1. TypeScript (Types,Auto complete,OOP, generics etc...)
2. Full-featured framework with well-tested defaults
3. Strong corporate support (Google)
4. Clear best practices
5. NativeScript, Iconic Framework
6. Two-way data binding
7. Google & Microsoft
8. Dependency injection

9. Templates, based on an extended version of HTML
10. Routing, provided by @angular/router
11. Ajax requests by @angular/http
12. @angular/forms for building forms
13. Component CSS encapsulation
14. XSS protection
15. Utilities for unit-testing components.
16. AngularCLI
17. Material Design Components

Disadvantages

1. Release frequency https://angular.io/guide/releases#release-frequency
2. Support policy and schedule
3. Learning curve can be steep
4. Two-way data binding
5. Complexity

Angular Resources

- A curated list of awesome Angular resources

Notes to consider with Angular

angular-vs-react-vs-vue-a-2017

A note on Vue

Vue is realized by a group of individuals whose work is supported via Patreon and other means of sponsorship. You can decide for yourself whether this is a positive or negative. Matthias Götzke thinks that Vue’s small team is a benefit because it leads to cleaner and less over-engineered code / API.