

- [HOME](#)
- [WORKS](#)
- [TECHNOLOGIES](#)
- [SERVICES](#)
- [COMPANY](#)
- [BLOG](#)
- [CONTACTS](#)
- [Tech](#)
- [Design](#)
- [Marketing](#)
- [Yalantis](#)

[SUBSCRIBE](#)

- [Yalantis](#)
- >
- [Blog](#)
- >
- [Tech](#)

Angular: Best Use Cases and Reasons To Opt For This Tool



Liliia Harkushko
Writer

- SHARE

0

-
-
-
-

Angular was ranked the second most used technology after Node.js in the [2018 Stack Overflow survey](#). The reasons for such a high rank are clear: this frontend tool is equipped with robust components to help developers write readable, maintainable, and easy-to-use code. But despite all the strengths of Angular, you should understand that sometimes this tool can work against you and your service rather than help it.

When choosing a technology stack for your project, it's important to pick the right tools that will speed up development and cut your costs. In this article, we explain when Angular is a must for your project and when you need to avoid it.

Angular overview

Angular (also referred to as Angular 2+) is a web development platform built in TypeScript that provides developers with robust tools for creating the client side of web applications.

Released in 2010 and formerly known as AngularJS, Angular is a JavaScript framework that was initially geared toward building single-page applications. At the time, the popularity of SPAs was growing rapidly, and so did the popularity of the AngularJS framework. But over time, the emergence of new tools like React and Vue, which offered wider functionality, made the Angular team think about improving their product.

Read also: [JavaScript Frameworks: How to Make Your Choice](#)

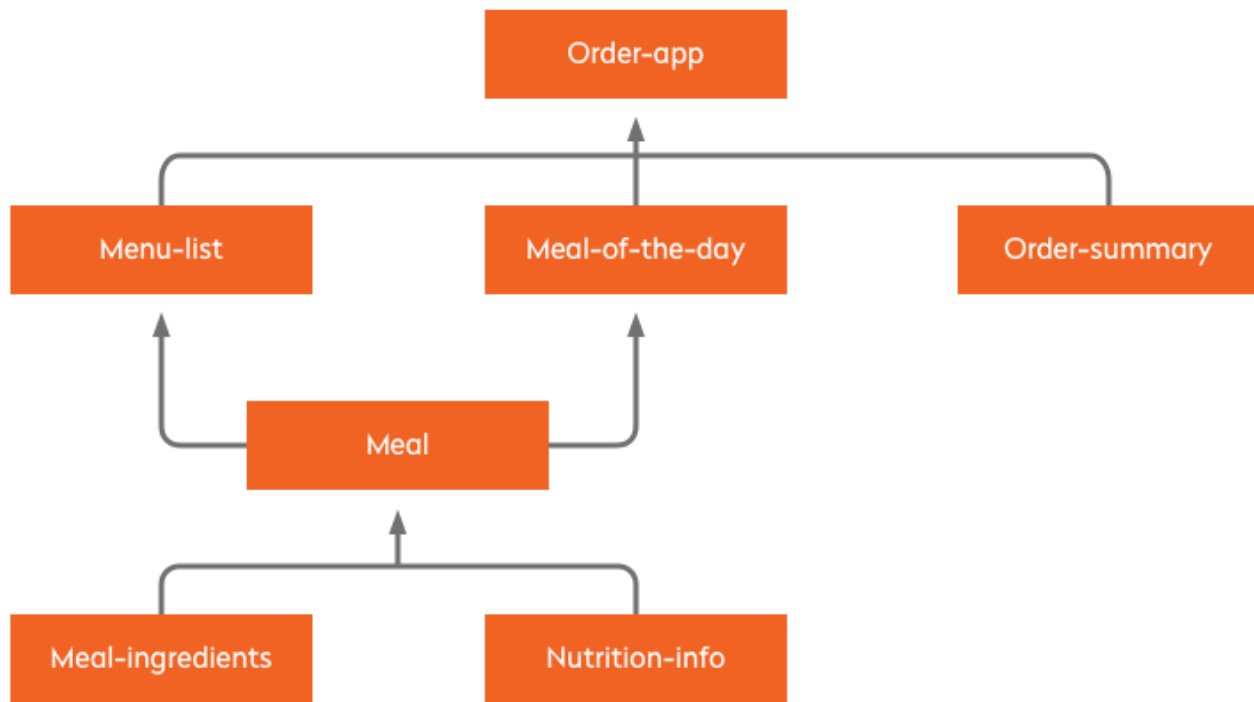
In 2016, Google presented a new, fully rewritten version of this tool and put it simply -- Angular. New features included semantic versioning based on the MAJOR.MINOR.PATCH scheme, a command-line interface (CLI), and an entirely component-based architecture. The major drawback was that this version was completely different from the previous and wasn't backward compatible. Developers had to rewrite all of their code if they wanted to use Angular instead of AngularJS in their projects.

The next version, Angular 4, became available in March 2017. This version offered faster compilation, reduction of code size, and better bug fix alerts and was backward compatible with Angular 2. A few months later, the Angular team released the next version of Angular with Build Optimizer, an improved compiler, and a dependency management update. Angular 6, released in May 2018, introduced the Ivy Renderer, new CLI commands, and Angular Elements. The next version of Angular is expected in late fall 2018. The constant improvement of this platform is the main reason why it's so beloved by frontend developers.

Why Angular?

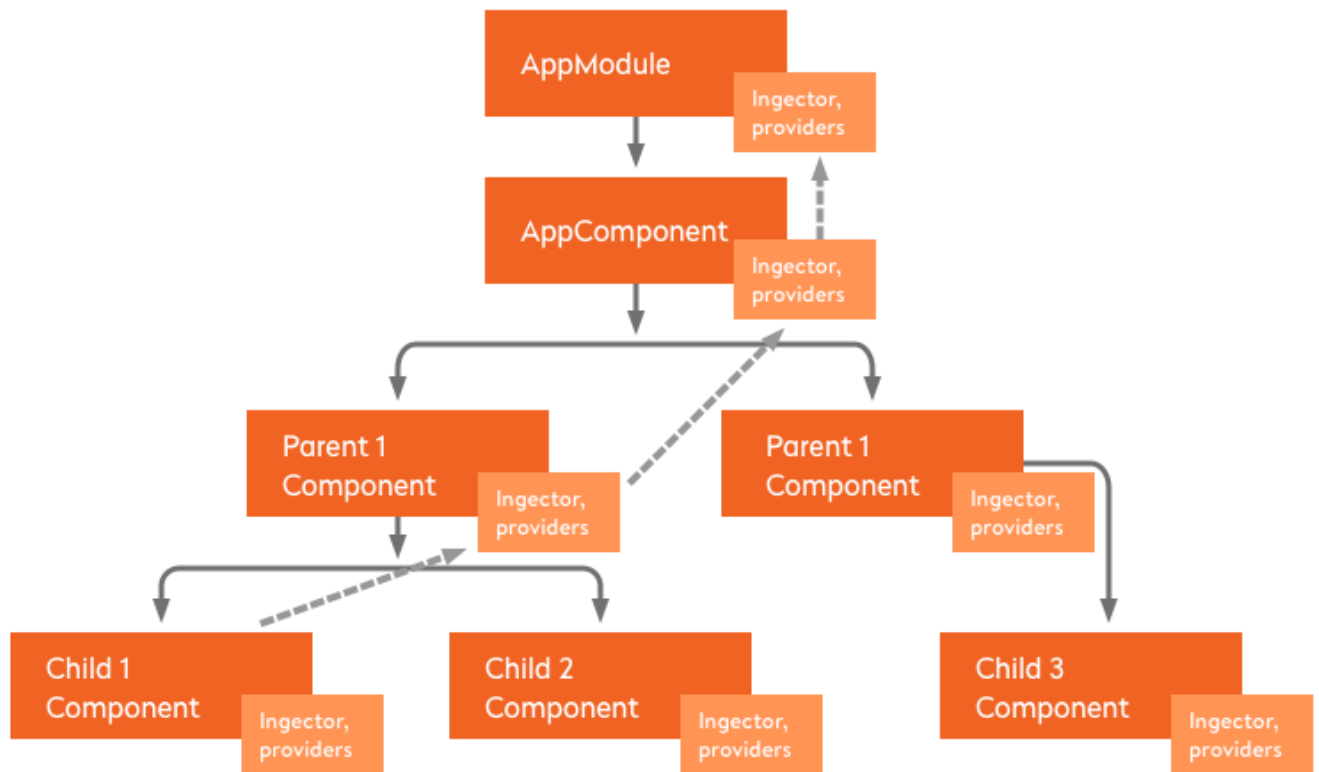
So, how developers can benefit from using Angular in their projects? Here we'll focus on Angular 2+, omitting AngularJS since we don't use the first version in our projects. The strong sides of Angular include:

- **Detailed documentation.** Angular boasts detailed documentation where developers can find all necessary information without asking their colleagues. As a result, developers can quickly come up with technical solutions and resolve emerging issues.
- **Support by Google.** A lot of developers consider Google support another benefit of Angular, making the platform trustworthy. At ng-conf 2017, the developers of Angular confirmed that Google will support Angular on a long-term basis.
- **Great ecosystem of third-party components.** The popularity of Angular has resulted in the appearance of thousands of additional tools and components that can be used in Angular apps. As a result, you can get additional functionality and productivity improvements.
- **Component-based architecture.** In the second version, Angular shifted from an MVC to a component-based architecture. According to this architecture, an app is divided into independent logical and functional components. These components can easily be replaced and decoupled as well as reused in other parts of an app. In addition, component independence makes it easy to test a web app and ensure that every component works seamlessly.



[Example of component-based architecture. Components of an order app]

- **Ahead-of-time compiler.** Angular's AOT compiler converts TypeScript and HTML into JavaScript during the build process. This means that code is compiled before the browser loads your web app so that it's rendered much faster. An AOT compiler is also much more secure than a just-in-time (JIT) compiler.
- **CLI.** It is probably the most beloved feature for the majority of Angular developers. It automates the whole development process making app initialization, configuration, and development as easy as possible. The Angular CLI allows you to create a new Angular project, add features to it, and run unit and end-to-end tests with a few simple commands. It not only increases code quality but also greatly facilitates development.
- **Angular Elements.** Starting with Angular 6, developers can easily add custom elements to any web app built with React, JQuery, Vue, or any other environment.
- **Ivy Renderer.** It's another great feature that became available with Angular 6. Ivy Renderer translates an app's components and templates into JavaScript code that can be displayed by a browser. The main characteristic of this tool is its tree shaking technique. During rendering, Ivy removes unused code to make it clearer and smaller. As a result, web apps load faster.
- **Angular Material.** This collection of Material Design elements optimized for Angular lets developers easily integrate UI components.
- **Dependency injection.** is quite an arguable advantage of Angular. In plain English, dependency injection refers to one object supplying the dependencies of another object. These dependencies define how various components are connected and show how changes in one part of the code affects other parts. On the one hand, using dependency injection makes code more readable and maintainable. It can greatly reduce the time spent testing and hence cut the costs of web development. Starting from version 2, Angular provides developers with a separate tree of dependency injectors that can be changed or replaced without reconfiguring all components. But on the other hand, dependency injection may be time-consuming and it may be hard to create dependencies for components.



[Example of dependency injections. The dotted arrow shows the dependency between Child1 Component and App Module]

- **Angular Universal.** Used for server-side rendering, Angular Universal brings several benefits to your project. First, it facilitates web crawlers to improve your web app's rank in search engines. Second, it decreases page load time and improves performance on mobile devices. These factors help you increase the number of users on your website.

Angular imperfections

Every tool has both strong and weak sides. To make a wise decision, you should also consider what difficulties you may face while using Angular. Angular has the following imperfections:

Hard to learn

Most developers claim that Angular is quite complex, which is why it can be difficult for newcomers. And even though the component-based architecture makes code more readable, it's still difficult to manage and set dependencies between components. TypeScript only adds fuel to the fire since you need additional time to learn it.

Poor SEO options

Even though the Angular team does their utmost to make Angular SEO-friendly, lots of developers still complain about poor accessibility for search crawlers. This is explained by the fact that single-page applications often change content and meta tags using JavaScript. Some search bots are unable to see these changes and hence see only the basic page template. Besides, deep links on a single-page app are hard to get indexed. Usually, developers need to make additional efforts to SEO optimize web apps built with Angular. Angular Universal and Prerender.io are common solutions.

Difficulties with backward compatibility

Developers can't switch directly from AngularJS to Angular. There's an entire section in the Angular documentation that enumerates all possible ways to deal with migration.

But it's fair to say that older versions of Angular are perfectly backward compatible. For example, you'll have no difficulty migrating from the fifth to the fourth version.

What projects can benefit from Angular?

1. Enterprise web apps

TypeScript has all the features needed to develop large-scale projects, as the Angular team claims. TypeScript is equipped with autocompletion, advanced refactoring, and navigation features. What's more, thanks to the architecture of this tool, you can easily reuse and maintain code.

2. Apps with dynamic content

Since the primary purpose of Angular was to create single-page web applications, it has a wide range of tools for SPA development. What's more, it's an ideal technology for websites where the content should change dynamically based on

user behavior and preferences. Dependency injections ensure that in case one component is changed, other components related to it will be changed automatically.

Read also: [Single-Page Apps vs Multi-Page Web Apps: What to Choose for Web Development](#)

3. Progressive web apps (PWAs)

[PWAs](#) were developed by Google in 2015. The Angular team foresees the upcoming popularity of PWAs and is working on facilitating the process of creating PWAs. Angular 5 is equipped with built-in PWA support. New CLI commands that came with Angular 6 enable developers to turn their web apps into progressive web apps easily.

What projects don't need Angular?

Sometimes Angular only burdens your project. We don't recommend using this tool for the following use cases:

- **Lightweight websites with static content.** Angular is ideal for websites with dynamic content, but it's insufficient when building small pages with static content. Implementing Angular in this case only increases the overall size of the project and hence increases the load time significantly. This is the main reason why we don't recommend you use Angular for landing pages and websites with static content.
- **SEO-optimized websites.** Angular is far from being SEO-friendly, so you should refrain from using Angular when your goal is to reach the top in search results. Otherwise, be ready to make additional efforts in order to optimize your web app for search engines.
- **Short-term projects.** Angular isn't a good solution for short-term projects or startups with limited resources. It's also not a good idea for projects with few developers, since the framework's complexity may be hard to overcome.
- **Apps with a microservice design approach.** Angular isn't suitable for applications with a microservice architecture on the frontend. If you want to be more flexible in terms of choosing additional tools, don't opt for this tool. Angular is a complete solution — it doesn't give you full control of the app size by letting you select only the things you really need.

Popular web apps built with Angular

The many benefits Angular provides to developers is the main reason why numerous popular companies around the globe use this tool for their projects. Below, we list some popular companies using Angular:



[These websites are built with Angular]

There are a couple of crucial things to consider when choosing whether to use Angular. When you think through the tech stack, you should carefully analyze your project. If you have any questions or want us to help you make the right decision, [write to us](#). We'll help you with all questions related to software development.

4.3/ 5.0

Article rating

209

Reviews

Remember those Facebook reactions? Well, we aren't Facebook but we love reactions too. They can give us valuable insights on how to improve what we're doing. Would you tell us how you feel about this article?

Wow!

Mmm

Hmm

Meh

Pff

[Get stories to your email every Thursday!](#)[Want stories to your inbox?](#)

[Insights](#)

[Single-Page Apps vs Multiple-Page Web Apps: What to Choose for Web Development](#)

[Insights](#)

[JavaScript Frameworks in 2019: How To Make Your Choice](#)

[Tech](#)

[How to Choose a Web Protocol: Comparison of SOAP, REST and GraphQL](#)

Discover what technologies we're good at

We use modern, proven technologies that allow us to effectively extend and scale our products

[Check our technology index](#)

Full-stack mobile (iOS, Android) and web app design and development company.

- [Works](#)
- [Technologies](#)
- [Services](#)
- [Company](#)
- [Blog](#)
- [Contacts](#)

-
-
-
-
-
-

[Stay tuned for our updates](#)

© Copyright 2019 Yalantis – Full-stack mobile (iOS, Android) and web app design and development company. All Rights Reserved

[Privacy Policy](#)

Nothing Found. Please try again with some different keywords.



Learn about digitalization with our newsletter

We'll send how-to articles, case studies, and company updates to your inbox every Thursday!

Enter your email address

☐ I accept your [Privacy Policy](#)

We use cookies to personalize our service and to improve your experience on the website and its subdomains. We also use this information for analytics.

[More info](#)