WHITE PAPER

What You Need to Know About AngularJS End of Life

TechnologyAdvice

Introduction

Businesses that are using open source software cannot rely on specific version numbers or predictable life cycles to gauge whether or not support will continue and for how long. AngularJS is no exception. After 11 years including a six-month extension because of the pandemic, AngularJS became end of life (EOL) on December 31, 2021.

About Angular and AngularJS

AngularJS is a popular framework for building JavaScript web applications. For developers, AngularJS combines clarity of templates, dependency injection, and end-to-end tooling. It helps handle cross-browser issues and generally speeds up development, making it a good choice for developers that need to move quickly to create web applications.

As the concept of development stacks took off, AngularJS became a common component. The popular MEAN stack consists of MongoDB, Express.js, AngularJS, and Node.js. Thousands of applications have been built with the MEAN stack. But today, developers can create their own stacks using different databases and a language like Node.js for the backend and Angular for the frontend, for example.

AngularJS, which corresponds to Angular version 1, started in 2010. It was written on JavaScript, and was not built with mobile support in mind. Angular 2 and all newer versions were written in Typescript. Angular (without the JS) is now built to support mobile experiences, a decision that was made because mobile was fast becoming the most popular method for consuming web applications.

The Current State of Angular Long Term Support

Prior to December 31, 2021, long-term community support for AngularJS included critical security patches for versions 1.6x and 1.8.x focused on major issues related to security and browser updates.

Long-term support meant the availability of fixes for:

- Major security flaws in the 1.8.x branch
- When a major browser releases a version that will cause current production applications using AngularJS 1.8.x to stop working

For Angular.io support (Angular 2 and newer), all of the major versions are supported for 18 months, which includes six months of active support and one year of long-term support. Angular versions 4, 5, 6, 7, 8 and 9 are no longer supported; long-term support end dates for Angular.io 10-12 are as follows:

- Version 10: Dec. 24, 2021
- Version 11: May 1, 2022
- Version 12: Nov. 12, 2022

The Risks of Unsupported EOL AngularJS

Ignoring the end of life for AngularJS carries similar risks to ignoring end of life for other open source projects. Those risks can include:

- Unpatched vulnerabilities: Unpatched CVEs will increase the risk of exploits.
- Incompatibility: Applications will no longer work with other components like web browsers.
- Compliance: Many internal IT compliance policies require a strategy for support.
- Complexity: Upgrades and migrations are more complex when they involve outdated software.

Sometimes using software that hasn't been updated can negatively impact application performance. As both hardware and software continue to evolve, unpatched software cannot keep pace. The libraries and components created to improve performance are not available for end of life software.

Self-support also carries costs. It requires time of developers and takes human resources away from new development projects. Self-support also means having in-house expertise in AngularJS, something that will be more difficult and more costly to find after end of life.

Businesses that decide to provide self-support for AngularJS are taking on the responsibility to monitor, fix, and patch security vulnerabilities. They also need to ensure interoperability with dependent components, as well as ensuring new web browser version compatibility. Self-support can also include hosting forks and dealing with backlogs, requests, and bugs, as well as managing an open source upstream community if the support is done in the open.

Options for AngularJS End of Life

With the above in mind, it's time for organizations to make a choice about the path forward. There are essentially four options for dealing with AngularJS EOL:

1. Keep your app on AngularJS and live with EOL risks.
2. Completely rewrite your app to the latest version of Angular.
3. Completely rewrite your app using another JavaScript framework like Vue.js, React.js, or Next.js.
4. Keep your app on AngularJS and work with a vendor to provide extended long-term support.

Get Extended LTS for AngularJS From OpenLogic

The fourth option mentioned above is to work with a vendor to extend the long-term support of AngularJS.

Extended AngularJS LTS support from OpenLogic offers long-term support until 2030 and includes security patches for medium and high severity vulnerabilities with guaranteed SLAs for CVSS score 7+, and fixes for web browser and jQuery updates that cause breaking changes. It's the best way to manage the security risks of end-of-life AngularJS.

TALK TO AN EXPERT

www.openlogic.com/solutions/angularjs-support-and-services