



Search



ABOUT ANGULAR >

INTRODUCTION

GETTING STARTED >

FUNDAMENTALS >

TECHNIQUES >

DEV WORKFLOW >

CONFIGURATION >

EXTENDING ANGULAR >

TUTORIALS >

RELEASE INFORMATION ✓

Keeping Up-to-Date

[Release Practices](#)

Updating to Version 10 >

Angular versioning and releases



Contents >

[Angular versioning](#)[Supported update paths](#)[Preview releases](#)

...

We recognize that you need stability from the Angular framework. Stability ensures that reusable components and libraries, tutorials, tools, and learned practices don't become obsolete unexpectedly. Stability is essential for the ecosystem around Angular to thrive.

We also share with you the desire for Angular to keep evolving. We strive to ensure that the foundation on top of which you are building is continuously improving and enabling you to stay up-to-date with the rest of the web ecosystem and your user needs.


This document contains the practices that we follow to provide you with a leading-edge app development platform, balanced with stability. We strive to ensure that future changes are always introduced in a predictable way. We want

• [Angular versioning and releases](#)

[Angular versioning](#)[Supported update paths](#)[Preview releases](#)[Release frequency](#)[Support policy and schedule](#)[Deprecation practices](#)[Public API surface](#)[Angular Labs](#)

later. If you are currently using AngularJS, see [Upgrading from AngularJS](#). *AngularJS* is the name for all v1.x versions of Angular.

Angular versioning

Angular version numbers indicate the level of changes that are introduced by the release. This use of [semantic versioning](#)  helps you understand the potential impact of updating to a new version.

Angular version numbers have three parts: `major.minor.patch`. For example, version 7.2.11 indicates major version 7, minor version 2, and patch level 11.

The version number is incremented based on the level of change included in the release.

- **Major releases** contain significant new features, some but minimal developer assistance is expected during the update. When updating to a new major release, you may need to run update scripts, refactor code, run additional tests, and learn new APIs.
- **Minor releases** contain new smaller features. Minor releases are fully backward-compatible; no developer assistance is expected during update, but you can optionally modify your apps and libraries to begin using new

expected during update.

Note: As of Angular version 7, the major versions of Angular core and the CLI are aligned. This means that in order to use the CLI as you develop an Angular app, the version of `@angular/core` and the CLI need to be the same.


Supported update paths

In alignment with the versioning scheme described above, we commit to support the following update paths:

- If you are updating within the **same major version**, then you can skip any intermediate versions and update directly to the targeted version. For example, you can update directly from 7.0.0 to 7.2.11.
- If you are updating from **one major version to another**, then we recommend that you **don't skip major versions**. Follow the instructions to incrementally update to the next major version, testing and validating at each step. For example, if you want to update from version 6.x.x to version 8.x.x, we recommend that you update to the latest 7.x.x release first. After successfully updating to 7.x.x, you can then update to 8.x.x.

We let you preview what's coming by providing "Next" and Release Candidates (rc) pre-releases for each major and minor release:

- **Next:** The release that is under active development and testing. The next release is indicated by a release tag appended with the `-next` identifier, such as `8.1.0-next.0`.
- **Release candidate:** A release that is feature complete and in final testing. A release candidate is indicated by a release tag appended with the `-rc` identifier, such as version `8.1.0-rc.0`.

The latest next or rc pre-release version of the documentation is available at next.angular.io .

Release frequency

We work toward a regular schedule of releases, so that you can plan and coordinate your updates with the continuing evolution of Angular.

Disclaimer: Dates are offered as general guidance and will be adjusted by us when necessary to ensure delivery of a high-quality platform.

- A patch release and pre-release (next or rc) build almost every week

This cadence of releases gives eager developers access to new features as soon as they are fully developed and pass through our code review and integration testing processes, while maintaining the stability and reliability of the platform for production users that prefer to receive features after they have been validated by Google and other developers that use the pre-release builds.

Support policy and schedule

All of our major releases are supported for 18 months.

- 6 months of *active support*, during which regularly-scheduled updates and patches are released.
- 12 months of *long-term support (LTS)*, during which only critical fixes and security patches are released.

The following table provides the status for Angular versions under support.

VERSION	STATUS	RELEASED	ACTIVE ENDS	LTS ENDS

		2020	2020	2021
^9.0.0	Active	Feb 06, 2020	Aug 06, 2020	Aug 06, 2021
^8.0.0	LTS	May 28, 2019	Nov 28, 2019	Nov 28, 2020

Angular versions ^4.0.0, ^5.0.0, ^6.0.0 and ^7.0.0 are no longer under support.

Deprecation practices

Sometimes "breaking changes", such as the removal of support for select APIs and features, are necessary to innovate and stay current with new best practices, changing dependencies, or changes in the (web) platform itself.

To make these transitions as easy as possible, we make these commitments to you:

- We work hard to minimize the number of breaking changes and to provide migration tools when possible.

- **Announcement:** We announce deprecated APIs and features in the [change log](#). Deprecated APIs appear in the [documentation](#) with ~~striketrough~~. When we announce a deprecation, we also announce a recommended update path. For convenience, [Deprecations](#) contains a summary of deprecated APIs and features.
- **Deprecation period:** When an API or a feature is deprecated, it will still be present in the next two major releases. After that, deprecated APIs and features will be candidates for removal. A deprecation can be announced in any release, but the removal of a deprecated API or feature will happen only in major release. Until a deprecated API or feature is removed, it will be maintained according to the LTS support policy, meaning that only critical and security issues will be fixed.
- **npm dependencies:** We only make npm dependency updates that require changes to your apps in a major release. In minor releases, we update peer dependencies by expanding the supported versions, but we do not require projects to update these dependencies until a future major version. This means that during minor Angular releases, npm dependency updates within Angular applications and libraries are optional.

Public API surface



Any changes to the public API surface will be done using the versioning, support, and depreciation policies describe above.

Angular Labs

Angular Labs is an initiative to cultivate new features and iterate on them quickly. Angular Labs provides a safe place for exploration and experimentation by the Angular team.

Angular Labs projects are not ready for production use, and no commitment is made to bring them to production. The policies and practices that are described in this document do not apply to Angular Labs projects.

RESOURCES

- [About](#)
- [Resource Listing](#)
- [Press Kit](#)
- [Blog](#)

HELP

- [Stack Overflow](#)
- [Gitter](#)
- [Report Issues](#)
- [Code of Conduct](#)

COMMUNITY

- [Events](#)
- [Meetups](#)
- [Twitter](#)
- [GitHub](#)

LANGUAGES

- [简体中文版](#)
- [正體中文版](#)
- [日本語版](#)
- [한국어](#)

Search



version 10.0.7 local build 1007 ac100.