Published in Angular Blog

Mark Thompson (@marktechson) Follow

Nov 4, 2021 · 7 min read · Listen

Angular v13 is now Available

We're back with the brand new release of Angular v13 to share with all of you! This latest release brings all sorts of updates and features to help your teams build great apps.

Get Angular v13 now by running `ng update` in your project. We also have an update guide available at update.angular.io to help teams get instructions on how to update their projects.

With each new release our goal is to find material ways to make Angular better. In this release, we've done that through expansion of Ivy-based features and optimizations, partnering with our excellent Angular community and continuing to provide a smooth, stable update process for your teams and projects.

Let's start with a look at how we're moving Angular forward with the power of Ivy.

Moving Angular rendering into the future

In the v12 release in May 2021 we talked about "Ivy Everywhere" and mentioned


landed some impactful changes in v13 as Ivy continues to open doors for optimizations and improvements.

State of View Engine

View Engine is no longer available in Angular as of v13. This is great news because Angular can continue to create Ivy-based features that bolster your productivity with the platform. Removing View Engine also means that Angular can reduce its reliance on `ngcc` (Angular compatibility compiler) in the future, and teams can look forward to faster compilation because metadata and summary files are no longer included.

Changes to the Angular Package Format (APF)

The Angular Package Format (APF) has been streamlined and modernized to better serve developers. To streamline the APF in v13 we've removed older output formats, including View Engine specific metadata.

To modernize it, we've standardized on more modern JS formats such as ES2020. Libraries built with the latest version of the APF will no longer require


We've also updated the APF to support Node Package Exports. This will help developers from inadvertently relying on internal APIs that may change.

Component API updates

Ivy also enables quality of life improvements to the way developers can dynamically create components. The API has now been simplified. Before the changes in Angular v13, dynamically creating components required a lot of boilerplate code.

The new API removes the need for `ComponentFactoryResolver` being injected into the constructor. Ivy creates the opportunity to instantiate the component with `ViewContainerRef.createComponent` without creating an associated factory.

Here's an example of creating components with previous versions of Angular:


With the new API, this code can become:

End of IE11 support

We heard your feedback and worked to pave a path forward with the removal of IE11 support in Angular v13.

Removing IE11 support allows Angular to leverage modern browser features such as CSS variables and web animations via native web APIs. What's more is that apps will be smaller and load faster because we can remove IE specific polyfills and code paths. It also removes the need for differential loading. Developers will benefit from improved APIs and build infrastructure while application users will benefit from faster loading and an improved user experience.

Running `ng update` will automatically drop these IE-specific polyfills and reduce bundle size during project migration.

Thanks to everyone who participated in the request for comments (RFC). Developers who still need to support IE11 users for existing projects can continue to use Angular v12 and it will be supported until November 2022.


On to the updates to Angular's tooling. Angular now supports the use of persistent build cache by *default for new v13 projects*. The valuable feedback from [RFC] Persistent build cache by default led to this tooling update that results in up to 68% improvement in build speed and more ergonomic options. In order for existing projects that have been upgrading to v13 to enable this features developers can add this configuration to `angular.json`:

Find out more details in the documentation.

ESBuild also sees some performance improvements in this release! We introduced `esbuild`, which now works with `terser` to optimize global scripts. In addition, `esbuild` supports CSS sourcemaps and can optimize global CSS, as well as optimizing all style sheets.

Framework changes and dependency updates

Angular v13 also features some helpful updates and important changes. First up, RxJS 7.4 is now the default for apps created with ... Existing apps using

command. To learn more about the changes from version 6 to version 7, check out this summary on rxjs.dev.

If that wasn't enough, there's now support for TypeScript 4.4. More information can be found by checking out the TypeScript release blog.

Improvements to Angular tests

We've made some important improvements to `TestBed` that now does a better job of tearing down test modules and environments after each test. The DOM is now cleaned after every test and developers can expect faster, less memory-intensive, less interdependent, and more optimized tests.

This feature has been opt-in since 12.1.0 and now it'll be the default while remaining customizable. Here's how - it can be configured for the entire test suite via the `TestBed.initTestEnvironment` method:


Or it can be configured per module by updating the

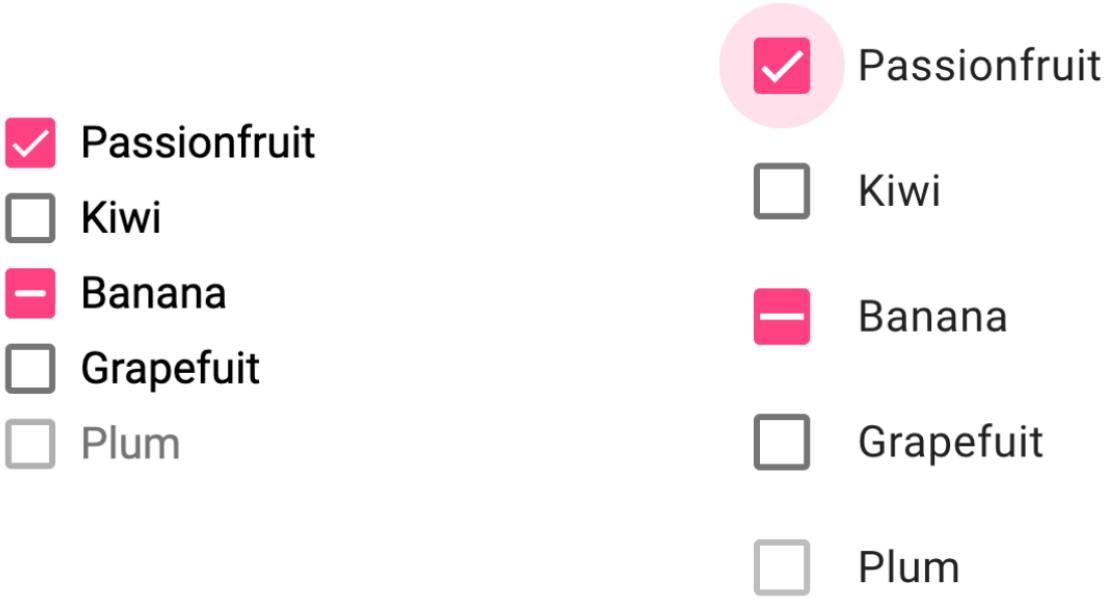

This provides the flexibility to apply these changes where they make the most sense for each project and its tests. Check out this blog by Lars Gyrup Brink Nielsen to learn even more.

All about components

Accessibility (a11y) has to be the foundation of everything we build inside and out of the Angular community. We take this responsibility seriously, and the work we've done has resulted in meaningful improvements and changes to Angular Material components.

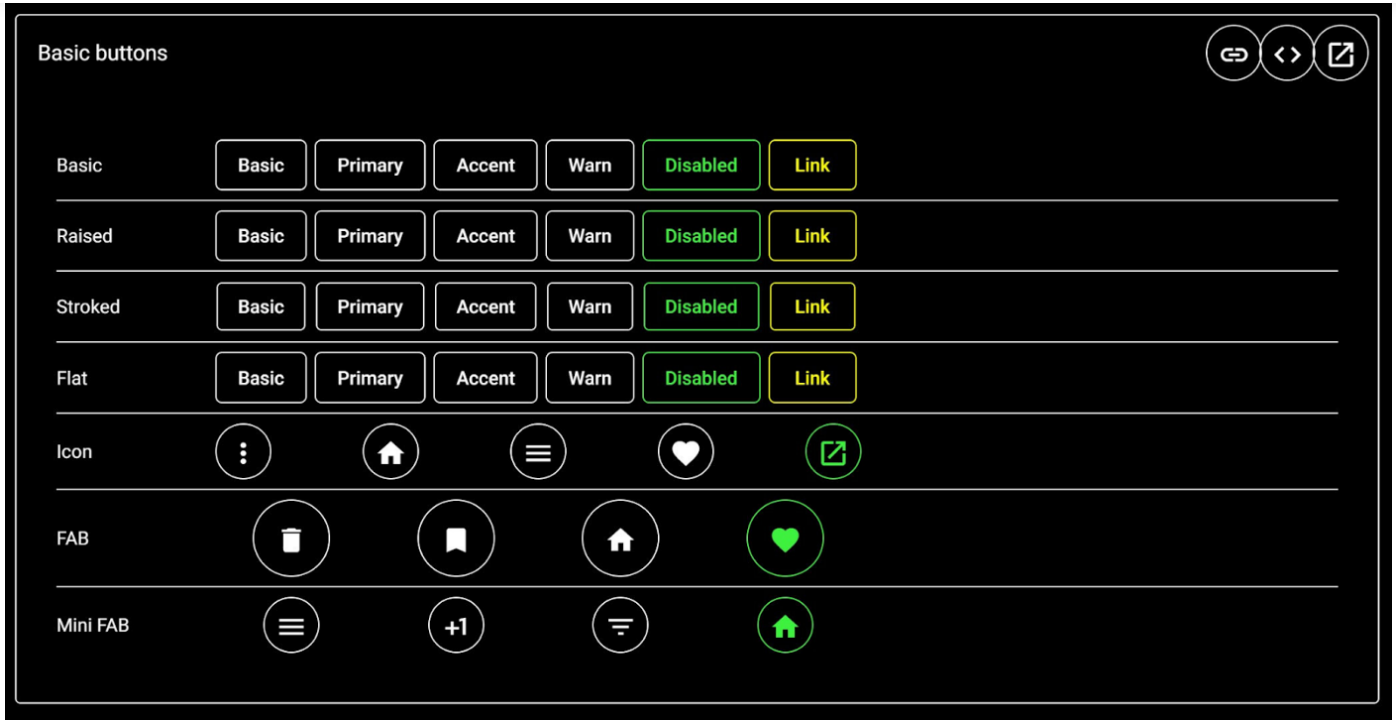
All the MDC-based components have been evaluated to meet elevated a11y standards on areas such as contrast, touch targets, ARIA, and more.

To get a better idea of how these changes impact components, have a look at the adjustments we've made to the touch target sizes for components like checkbox and radio button.

A comparison of touch target sizes. The sizes on the right are the new sizes.

There have also been some improvements to high contrast modes for multiple components.



Material Components in high contrast mode

Learn more about these changes in our blog post on improving Angular Component's accessibility. We hope this helps everyone to build more inclusive Angular applications.

Other notable updates

fonts can improve your app performance by speeding up the First Contentful Paint (FCP). This change is now enabled for everyone by default! All you need to do is `ng update`. We have a video on font inlining that can be helpful, check it out here:


feature smaller sections to make the learning journey clearer. We've also added more API documentation for `$localize`.

Community contributions

The Angular community never ceases to show up in a huge way by adding features to the framework. The Angular team is incredibly grateful for such a vibrant, supportive community. Let's take a moment to highlight ***a few*** of the contributions that made it into this release.

Dynamically enable/disable validators

Submitted by Nirmal Bhagwani, this PR allows built-in validators to be disabled by setting the value to `null`. This becomes increasingly helpful when building dynamic forms.

Restore history after canceled navigation

Ahmed Ayed contributed a PR that allows the `canceledNavigationResolution`


These are a few of the updates we're highlighting, but even more contributions came from the community:

- Making the SwUpdate API a little more ergonomic
- Language Service config to enable auto-apply optional chaining on nullable symbol
- Router emit activate/deactivate events when an outlet gets attached/detached
- And more!

Huge shout out to all of you in the community who have made contributions to the framework. We move Angular forward together.

Closing thoughts

Angular continues to move forward with the help of the stellar Angular community. Thank you all for your contributions to the repository. Also, thank you for your invaluable feedback in the RFCs. Your support helps us shape the


For a more detailed overview check our full changelog. Get the latest version of Angular and let us know what you think; you can find a detailed update guide at update.angular.io.

Until the next time, friends, go build great apps.