

Angular 4.0.0 Now Available

What's New

Smaller & Faster

In this release we deliver on our promise to make Angular applications smaller and faster. By no means are we done yet, and you'll see us being focused on making further improvements in the coming months.

View Engine

We've made changes under the hood to what AOT generated code looks like. These changes reduce the size of the generated code for your components by around 60% in most cases. The more complex your templates are, the higher the savings.

During our release candidate period, we heard from many developers that migrating to 4 reduced their production bundles by hundreds of kilobytes.

Read the [Design Doc](#) to learn more about what we did with the View Engine.

Animation Package

We have pulled animations out of `@angular/core` and into their own package. This means that if you don't use animations, this extra code will not end up in your production bundles.

This change also allows you to more easily find documentation and to take better advantage of autocompletion. You can add animations yourself to your main NgModule by importing `BrowserAnimationsModule` from `@angular/platform-browser/animations`.

New Features

Improved `*ngIf` and `*ngFor`

Our template binding syntax now supports a couple helpful changes. You can now use an if/else style syntax, and assign local variables such as when unrolling an observable.

```
<div *ngIf="userList | async as users; else loading">
  <user-profile *ngFor="let user of users; count as count; index as i" [user]="user">
    User {{i}} of {{count}}
  </user-profile>
</div>
<ng-template #loading>Loading...</ng-template>
```

Angular Universal

Universal, the project that allows developers to run Angular on a server, is now up to date with Angular again, and this is the first release since Universal, originally a community-driven project, was adopted by the Angular team. This release now includes the results of the internal and external work from the Universal team over the last few months. The majority of the Universal code is now located in `@angular/platform-server`.

To learn more about taking advantage of Angular Universal, take a look at the new [renderModuleFactory](#) method in `@angular/platform-server`, or Rob Wormald's [Demo Repository](#). More documentation and code samples are forthcoming.

TypeScript 2.1 and 2.2 compatibility

We've updated Angular to a more recent version of TypeScript. This will improve the speed of ngc and you will get better type checking throughout your application.

Source Maps for Templates

Now when there is an error caused by something in one of your templates, we generate source maps that give a meaningful context in terms of the original template.

Packaging Changes

Flat ES Modules (Flat ESM / FESM)

We now ship flattened versions of our modules ("rolled up" version of our code in the EcmaScript Module format, see [example file](#)). This format should help tree-shaking, help reduce the size of your generated bundles, and speed up build, transpilation, and loading in the browser in certain scenarios.

Read more about the importance of Flat ES Modules in ["The cost of small modules"](#).

Experimental ES2015 Builds

We now also ship our packages in the ES2015 Flat ESM format. This option is experimental and opt-in. Developers have reported up to 7% bundle size savings when combining these packages with Rollup. To try out these new packages, configure your build toolchain to resolve ["es2015" property](#) in package.json over the regular ["module" property](#).

Experimental Closure Compatibility

All of our code now has Closure annotations, making it possible to take advantage of advanced Closure optimizations, resulting in smaller bundle sizes and better tree shaking.

Updating to 4.0.0

Updating to 4 is as easy as updating your Angular dependencies to the latest version, and double checking if you want animations. This will work for most use cases.

On Linux/Mac:

```
npm install @angular/{common,compiler,compiler-cli,core,forms,http,platform-browser,platform-browser-dynamic,platform-server,router,animations}@latest typescript@latest --save
```

On Windows:

```
npm install @angular/common@latest @angular/compiler@latest  
@angular/compiler-cli@latest @angular/core@latest @angular/forms@latest  
@angular/http@latest @angular/platform-browser@latest @angular/platform-
```

```
browser-dynamic@latest @angular/platform-server@latest  
@angular/router@latest @angular/animations@latest typescript@latest --save
```

Then run whatever `ng serve` or `npm start` command you normally use, and everything should work.

If you rely on Animations, import the new `BrowserAnimationsModule` from `@angular/platform-browser/animations` in your root `NgModule`. Without this, your code will compile and run, but animations will trigger an error. Imports from `@angular/core` were deprecated, use imports from the new package `import { trigger, state, style, transition, animate } from '@angular/animations';`.