-----------<https://blog.angular.io/version-9-of-angular-now-available-project-ivy-has-arrived-23c97b63cfa3>----------

**Version 9 of Angular Now Available — Project Ivy has arrived!**

How to update to version 9

First, update to the latest version of 8 -> **ng update @angular/cli@8 @angular/core@8**

Then, update to 9 -> **ng update @angular/cli @angular/core**

**-----Ivy-----**

*Version 9 moves all applications to use the Ivy compiler and runtime by default.*

**-Advantages-**

* Smaller bundle sizes
* Faster testing
* Better debugging
* Improved CSS class and style binding
* Improved type checking
* Improved build errors
* Improved build times, enabling AOT on by default
* Improved Internationalization

**-Smaller bundle sizes-**

The Ivy compiler has been designed to remove parts of Angular that aren’t being used via tree-shaking and to generate less code for each Angular component.

Small apps that don’t use many Angular features can benefit most from tree-shaking.

Large apps with many components can benefit most from the reduced factory size.

Medium-sized apps should see bundle sizes that are on par or slightly smaller, since they benefit less from tree-shaking and don’t have enough components to truly leverage smaller factories.

**-Faster testing-**

We have also revamped the implementation of TestBed in Ivy to make it more efficient.

Previously, TestBed would recompile all components between the running of each test, regardless of whether there were any changes made to components (for example, through overrides).

In Ivy, TestBed doesn’t recompile components between tests unless a component has been manually overridden, which allows it to avoid recompilation between the grand majority of tests.

With this change, the framework’s core acceptance tests are about 40% faster.

**-Better debugging-**

Ivy provides you with more tools to debug your applications. When running an application in Dev Mode with the Ivy runtime, we now offer the new ng

You can ask Angular for access to instances of your components, directives, and more

You can manually call methods and update state

When you want to see the results of change detection, you can trigger change detection with applyChanges

Ivy also improves the stack trace for debugging issues such as the ExpressionChangedAfterItHasBeenCheckedError. Previously the stack trace could be unhelpful:

With Ivy, you see a more useful stack trace that allows you to jump directly to the template instruction with the expression that has changed.

**-Improved CSS class and style binding-**

The Ivy compiler and runtime provides improvements for handling styles. Previously, if an application contained competing definitions for a style, those styles would destructively replace each other. With Ivy, the styles are merged in a predictable way.

<my-component style="color:red;" [style.color]="myColor" [style]="{color: myOtherColor}" myDirective></div>

@Component({ host: { style: "color:blue" },...})...@Directive({ host: { style: "color:black", "[style.color]": "property" },...})

Previously, whichever binding was evaluated last would win, and this could depend on the timing of changes to these expressions. If myColor and myOtherColor both were undefined, the static ‘red’ style would be ignored.

With version 9, you can manage your styles through a clear, consistent order of precedence that isn’t dependent on timing. The most specific styles always have the highest precedence. For example, a binding to [style.color] overrides a conflicting binding to [style].

**-Improved type checking-**

The Angular compiler can check more of the types of your application, and it can apply more strict rules. These features will help you and your team catch bugs earlier in the development process.

We support two main flags for additional type checks in addition to the default:

-fullTemplateTypeCheck — Activating this flag tells the compiler to check everything within your template (ngIf, ngFor, ng-template, etc)

-strictTemplates — Activating this flag will apply the strictest Type System rules for type checking.

**-Improved build errors-**

makes all of the error messages easier to read.

-Improved build times, enabling Ahead-of-Time compiler by default-

overhead decreased from 0.8x to 0.5x with Ivy, an improvement of nearly 40%.

These improvements mean that AOT builds can be noticeably faster.

This means that `ng serve` now benefits from the same compile-time checking as production builds, significantly improving the developer experience for Angular.

we also no longer require entryComponents. These components will be discovered and compiled automatically by their usage.

**-Improved internationalization (i18n)-**

In 9.0, we’re making this faster by moving the build-time i18n substitutions later in the build process. This change allowed us to make it up to 10 times faster.

-More reliable ng update-

Always use the latest CLI. Starting with 8.3.19 of the CLI, we now use the CLI from the destination version during updates. This means that going forward, updates will take advantage of newer update features automatically.

Clearer progress updates. ng update now does more to tell you what is going on under the hood. For each migration, you’ll see information about the migration.

Easier update debugging. By default, ng update runs all of the migrations and leaves the aggregate changes on disk for you to inspect. The version 9 update also introduces the new --create-commits flag. When you run ng update --create-commits, the tool commits the state of your codebase after each migration, so you can step through and understand or debug the changes we are making to your code.

**-New options for 'providedIn'-**

Two additional options.

platform— Specifying providedIn: 'platform' makes the service available in a special singleton platform injector that is shared by all applications on the page.

any— Provides a unique instance in every module (including lazy modules) that injects the token.

**-Component harnesses-**

Testing components has historically relied on using implementation details such as CSS selectors to find components and to trigger events. This meant that whenever a component library changed its implementation, all of the tests relying on those components would need to be updated.

In version 9, we are introducing component harnesses, which offer an alternative way to test components. By abstracting away the implementation details, you can make sure your unit tests are correctly scoped and less brittle.

-IDE & language service improvements-

TextMate grammar for Angular Template Syntax, which now enables syntax highlighting in both inline and external templates

“Go to definition” for templateUrl and styleUrls

NgModule and type information in hover tooltip

**-TypeScript 3.7 support-**

Angular has been updated to work with TypeScript 3.6 and 3.7, including the extremely popular optional chaining feature in TypeScript 3.7. To stay in sync with the ecosystem, we’ve also updated our version of other ecosystem dependencies such as Zone.JS and RxJS.