## Angular 7 highlights

Though there are not a lot of new updates & features launched in this release, but there are certainly some useful upgrades for Angular front-end developersincluding improvements to Angular Material and the core framework, CLI with synchronized major versions, and upgrades to the toolchain.

The Angular JS 7 version is primarily focused on the Ivy project, which has been going on since past release.

The Ivy project is basically rewriting the Angular compiler and runtime code to make it better, faster, and smaller.

While the Ivy is still not ready for the prime time yet, but back in April, there were a lot of rumors & fake news about Ivy as well as many other Angular features, which are now nowhere to be seen.

Let’s also take a look at some of those fake news and rumors that was intended as an April fool joke back then.

## Fake news & rumors about Angular JS 7

Below are the biggest rumors that were shared across the web in April, mainly.

* **@aiStore** – This was supposed to be an artificial intelligence storage, having the capability to be backwards compatible with previously created ngrx code base.
* **@angular/core splitting** – An answer on Quora was responsible for spreading the rumors of @angular/core splitting package, which was supposed to reduce an app bundle size and cold start performance.
* **New ng-compiler** – Though we couldn’t find the original source of this rumor like the first two, but it was expected that application will have a massive reduction in overall size of bundles, shockingly between 95-99%.

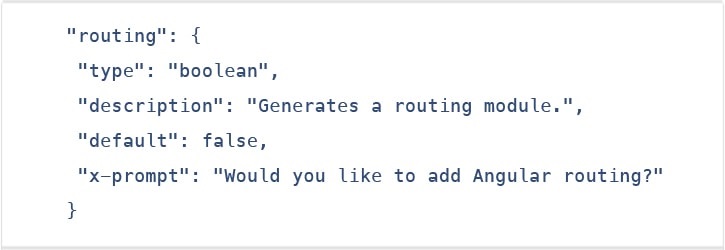
These were the main fake features that were rumored in April. So, if you see them in other articles, know that they are not real. Also, please spread the word as well!

## New features in angular 7

Now, without any further adieu, let’s dive into discussing the new features of Angular 7 one-by-one, and see how each of them improve the frontend development process for the better.

### 1. CLI prompts

In Angular 7, the CLI prompts have been updated to v7.0.2 with new features. For instance, it will now prompt users when typing commands like @angular/material, ng-new, and ng-add to help them discover the in-built SCSS support, routing, and more.



These CLI prompts, in addition, have been added to Schematics, so that all package publishing schematics can now benefit from CLI prompts.

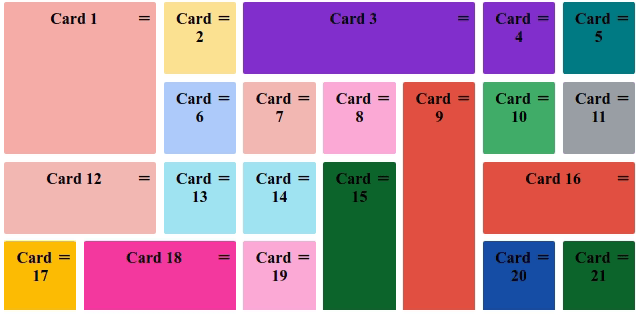
### 2. Angular material & component dev kit (CDK)

The Angular 7 introduced minor visual updates & improvements in Material Design that earlier received a major update this year only.

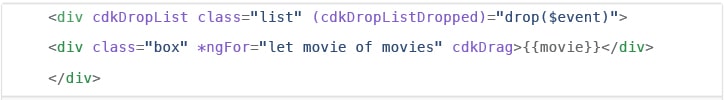
In addition, refresh, virtual scrolling, large lists of data, dynamic loading and unloading of parts of the DOM also were the part of improvements in CDK and Angular Material.

### 3. Drag & drop

The new drag-drop module basically provides a better way to easily create drag & drop interfaces, which is backed by sorting within a list, support for free dragging, animations, custom drag handles, transferring items between lists, previews, and placeholders.



In simple terms, the Drag-and-Drop support has now been implemented in CDK and it also includes automatic rendering as the users relocates items.





### 4. Virtual scrolling

Like mentioned earlier, the new Virtual Scrolling in Angular 7 basically loads and unloads items from the DOM depending upon visible parts of lists, resulting into a much faster experiences for users having huge scrollable lists.

This virtual scrolling package basically provides helpers, which react to all scroll events.



Simply put, it activates a high-performant way by making the height of container element exactly same as the height of total number of remaining elements to be rendered.

This, in turn, then renders the only visible items in view, creating faster experiences for the end-users.

### 5. Application performance improvements

The development team at Google have always focused on the performance improvements, and while doing so, they recently found that most of the developers were using reflect-metadata in their production, which actually was only required in the development.



So, to fix this problem, they’ve made a part of Angular JS 7 to automatically remove this from the polyfills.ts file.



### 6. No Ivy

No Ivy rendered in the Angular 7 according to the official information. The AngularJS development company’s team have said that the Ivy is in the pipeline; however, they haven’t disclosed its final timeline.

The official blog post also mentioned that the backward compatibility validation has begun. And its full beta version is expected to launch with Angular 8 version.

### 7. Documentation updates

Another key improvement introduced in the Angular 7 is of the documentation update.

The team has worked hard on improving the reference material and the guidelines for the betterment and convenience of the developers.

The documentation updates for Angular is an important step for the Angular CLI.

### 8. Dependency updates

Documentation are not the only things that have been updated. Even the dependencies have undergone upgradation for the third-party projects.

The support for Node 10, TypeScript3.1, and the RxJS6.3 all are included under this update.

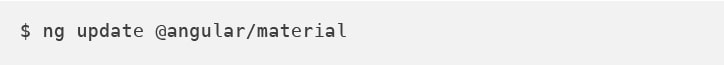
However, you would continue to receive the support if you already have Node 8. Talking about the latest update of TypeScript 3.1, it has now become compulsory for Angular 7 users to bump to TS 3.1.

## Upgrading requirements

The Upgrading process is really simple. For most Angular apps out there that are running on Angular 6 and RxJS 6, you only need to run a single command stated below to upgrade to Angular 7.



If, however, you’re using Angular Material, use following single line command and you’re good to go!



During the upgrade process, if anything seems unusual, check out the update guide to see if you need to execute any special commands.

## Takeaway

All in all, the Angular 7 seems like a far better solution. The Angular development team has certainly did a great job in making the Angular platform better.

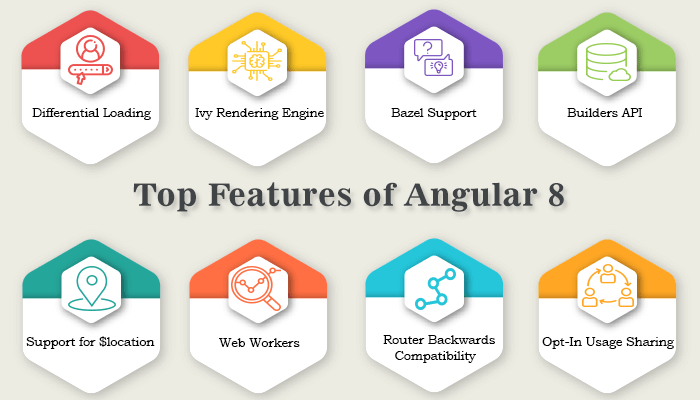
Therefore, if you’re planning to implement Angular in your own project, whether mobile or web, we recommend to hire AngularJS developer who is well-versed with the framework.

**Features Of Angular 8**

## What’s New in Angular 8

Angular 8 has arrived and with it a bunch of workflow and a new set of powerful features that Angular developers will appreciate like the core framework, Angular Material library, and the Command Line Interface. They have enabled major launch partner as Angular Console for running Angular projects, #angular/fire for integrating Firebase with Angular, StackBlitz integrated IDE and NativeScript for building native mobile apps with Angular. let’s go over and summarize what are new features introduced in Angular 8 as well as how upgrading your Angular 7 apps to Angular 8.

## List of Top 10 Angular 8 Features



### Preview of Ivy

If you have been following Angular then you probably encountered the word “Ivy”. Ivy is a major change in Angular history, it an angular renderer which is radically different from anything as it uses incremental DOM. It changes how the framework internally works, without changing our Angular applications. The Ivy project is basically rewriting the Angular compiler and runtime code in order to reach

* better build times ( incremental approach)
* better build sizes more compatible with tree-shaking
* new potential features like lazy loading of component instead of modules

#### How Angular Ivy works on Incremental DOM

The key idea behind Incremental DOM is every component gets compiled into a series of instructions. These instructions create DOM trees and update them in-place when the data changes.

@Component({  
 selector: 'todos-cmp',  
 template: `  
 <div \*ngFor="let t of todos|async">  
 {{t.description}}  
 </div>  
 `  
})  
class TodosComponent {  
 todos: Observable<Todo[]> = this.store.pipe(select('todos'));  
 constructor(private store: Store<AppState>) {}  
}

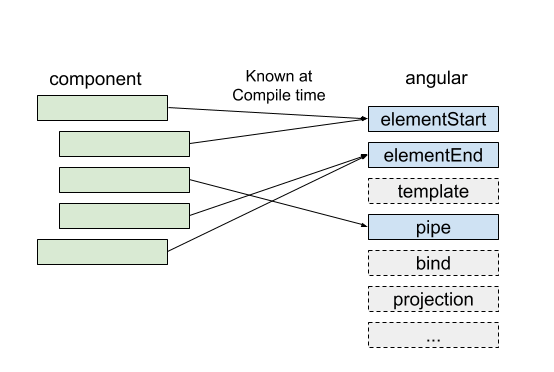
Will be compiled into:

var TodosComponent = /\*\* @class \*/ (function () {  
 function TodosComponent(store) {  
 this.store = store;  
 this.todos = this.store.pipe(select('todos'));  
 }  
 TodosComponent.ngComponentDef = defineComponent({  
 type: TodosComponent,  
 selectors: [["todos-cmp"]],  
 factory: function TodosComponent\_Factory(t) {  
 return new (t || TodosComponent)(directiveInject(Store));  
 },  
 consts: 2,  
 vars: 3,  
 template: function TodosComponent\_Template(rf, ctx) {  
 if (rf & 1) { /\*\* create dom\*/   
 pipe(1, "async");  
 template(0, TodosComponent\_div\_Template\_0, 2, 1, null, \_c0);  
 } if (rf & 2) { /\*\* create dom\*/   
 elementProperty(0, "ngForOf", bind(pipeBind1(1, 1, ctx.todos)));  
 }  
 },  
 encapsulation: 2  
 });  
   
 return TodosComponent;  
}());

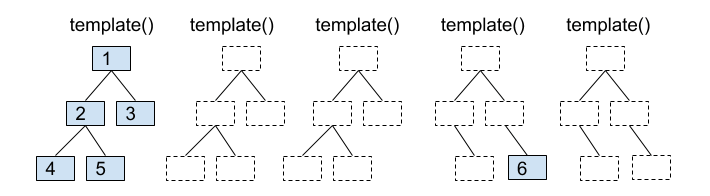
The template function contains the instructions rendering and updating the DOM as they are rendering engine.

**Two main concepts of IVY**

1. **Tree shakable**: It means removing unused pieces of your code, the framework does not interpret the component. Instead, the component references instructions. If it doesn’t reference a particular instruction, which will never be used so we can omit the unused instruction from the bundle results in smaller bundles and faster load times.

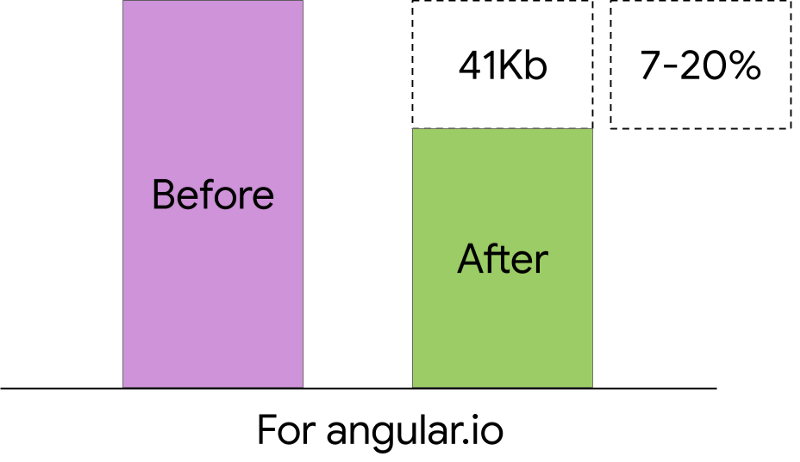


1. **Low Memory Footprint**: Incremental DOM doesn’t need any memory to rerender the view if it doesn’t change the DOM so it allocates the memory when the DOM nodes are added or removed. since most of render/template calls don’t change anything result in huge memory savings.



#### Differential loading

Angular 8 apps will now be more performant, thanks to Differential Loading of Modern JavaScript. With differential loading, new apps generated by Angular CLI will now contain separate bundles for legacy JavaScript (ES5) and modern JavaScript (ES2015+).



The correct bundle will be loaded automatically by the browser & will be able to download smaller, more efficient app bundles that load and render faster.

#### ****Angular Router Backwards Compatibility****

Angular 8 feature added backward compatibility mode to Angular router that helps to upgrade the path for large projects and will make it easier to move to Angular by allowing lazy loading parts of Angular v1.x apps using $route APIs.In simple word, we will be able to upgrade our Angular 1.x apps to Angular 2+ right away.

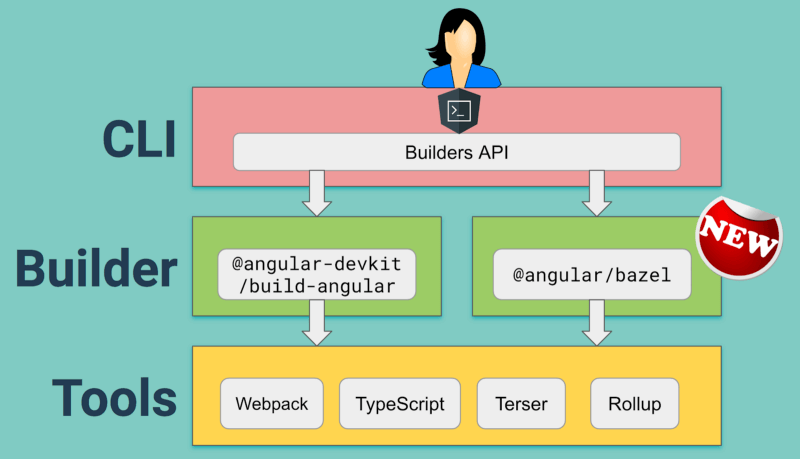
#### Improved Web Worker Bundling

Thanks to Angular CLI 8, web workers are included while building the production bundles which are essential for improving the parallelizability and helps increase the performance. Angular 8.0 is thus adding building support to CLI which provides one bundle for every web worker.

ng g webWorker <name>

#### Bazel Support

One of the new features of Angular 8 is the possibility to build your CLI application with Bazel. The Angular framework itself is built with Bazel.  
It is available as opt-in, it is expected to be included in @angular/cli in Version 9.



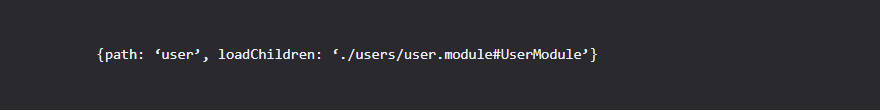
**Bazel key advantages are:**

* We will be able to build our backends and frontends with the same tool
* Faster build time The first build will be painfully slow, as Bazel is aiming for exactly reproducible builds but concurrent builds will be a lot faster & it will be beneficial if your app uses several modules and libraries.
* Incremental Build: Codebase will only trigger the smallest rebuild possible help to build and deploy only what has changed rather than the entire App.
* You can eject the Bazel files, they are hidden by default.
* The possibility to have remote builds (with cache) on a build farm

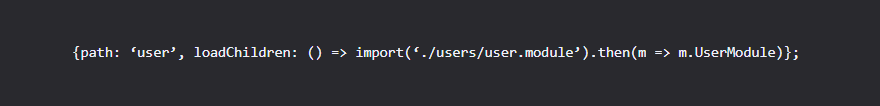
#### ****Lazy Loading****

Lazy loading is based on the concepts of Angular Routing as it helps bring down the size of large files by lazily loading the files that are required. In previous angular versions, the route configuration uses the property @loadChildren which accepts a string and if there was a wrong module name or any typo while writing the code, Angular would not consider it wrong and accept whatever value was there as a string until we try building it.

So to overcome that they have added support for dynamic imports in router configuration in latest Angular 8 which enable the use of import statement for lazy loading the module and this will be understood by the IDEs, webpack etc.



Now Editor will understand what's this syntax and will recognize if there is some mistake and we won't have to wait till build time to realize about an error.



#### ****Opt-In Usage Sharing****

To keep their efforts in alignment with the community’s needs Angular CLI will be gaining another new feature i,e opt-in usage sharing. This feature will enable opt-in to sharing telemetry about your Angular CLI usage with the Angular team and now Angular 8 can collect data like commands used and the build speed if users allow them which will help developers improve in future.

#### CLI workflow improvements

The Angular CLI is continuously improving, and now the ng-build, ng-test and ng-run are equipped to be extended by 3rd-party libraries and tool.   
For Example: **AngularFire**

#### ****Builders API****

The new version allows us to use Builders API. It uses builders for main operations like: serve, build, test, lint and e2e. Basically, the builder Builders are functions that implement the logic and behavior for a task that can replace a command which you pass to the createBuilder() method from @angular-devkit/architect package & now we can create our custom builders as well.

import { BuilderOutput, createBuilder } from '@angular-devkit/architect';  
export default createBuilder((options, context) => {  
 return new Promise<BuilderOutput>(resolve => {  
 resolve({ success: true });  
 });  
});

#### ****AngularJS API Migration Improvements**** with****$location service****

The Angular Team wants to provide support for all developers using AngularJS to upgrade them with latest Angular so some enhancements have been made to provide better integration with the AngularJS $location service in hybrid (AngularJS <=> Angular) apps. A new package angular/common/upgrade is added help you

* To retrieve the state from location service.
* To track all location changes.
* Help you retrieve hostname protocol port search properties which were available in AngularJS.
* MockPlatformLocation API added to test the location service.

#### ****Service Worker****

Angular ships with a service worker implementation Starting with version 5. With the Angular Service Worker and the Angular CLI built-in PWA support angular developers can take advantage of this service worker and benefit from the increased reliability and performance it provides, without needing to code against low-level APIs and can achieve native-like application download and Installation.

#### Updated Typescript to 3.4.x

In the newest version of Angular 8, they have updated Angular’s dependencies which include tools like RxJS and TypeScript to v3.4 ( Angular 7 uses v3.2) and new apps generated via the Angular CLI will also use the newest version of TypeScript by default.

#### ****Angular Firebase****

Angular have officially added support for firebase and now we can deploy our application using the Angular CLI.

ng run [PROJECT\_NAME]:deploy

#### Deprecated APIs and Features

Angular aims to balance innovation and stability in their framework and to do that they have removed or replaced some features & API'S so that Angular can stay updated with latest practices, changing dependencies, or changes in the platform itself.

To make these transitions easy they deprecate APIs and features for a period of time before removing them which provide developers time to update your apps to the latest APIs and best practices.

* Web Tracing Framework integration
* @angular/platform-webworker and@angular/platform-webworker-dynamic both the packages are deprecated
* Usage for any in TesBed.get
* Removed deprecated DOCUMENT from @angular/platform-browser
* @angular/http removed from the list of packages
* ngForm element selector
* Service worker versionedFiles

#### Angular Performance & Upgradtion from Angular 7 to Angular 8

Angular 8 new features are nice, but the main reason for many of us to upgrade to new versions of Angular 8 is to get a performance boost. If you worked with previous angular versions then upgrading an app from Angular 7 over to Angular 8 is simple.  
  
For most developers, one command should take care of this update :

ng update @angular/cli @angular/core <name>