

# Upgrading from AngularJS to Angular: A Complete Guide



Jemin Desai

Published on: April 10, 2024



## Quick Overview:

Angular is the knight in shining armor for many business owners worldwide. Not only the new entrepreneurs but also the established enterprises have chosen Angular for their web application development. However, the talk of the town is the road ahead for the business owners to take down the dilemma that post-AngularJS has now reached EoL status, so should the business owners shift to another framework or stay with Angular? In this blog post, we will understand if upgrading AngularJS to

Angular is the right call for your business. Also, we will glimpse the benefits, challenges, and steps by which you can migrate to Angular easily.

## Table of Contents



Introduction

AngularJS and Angular – Comparison Table

Why Migrate AngularJS to Angular?

1. Community and Ecosystem
2. Timely Updates
3. Third-Party Compatibility

Confused! If Migrating AngularJS To Angular Is The Right Choice For You?

1. Improved Performance

Setting Up the AngularJS to Angular Migration

1. Plan Your Migration Strategy
2. Create a New Build
3. Use the ngUpgrade Toolkit Library
4. Transfer Services and Components
5. Use a UI-router
6. The Ideal Path For Migration

Tools You Can Use To Migrate From AngularJS to Angular

1. NgUpgrade
2. Angular Upgrade Guide
3. Angular CLI
4. TSLint and Codelyzer
5. Manual Conversion Tools

AngularJS to Angular Challenges

1. Project Scale and Compatibility
2. Command Line Interface
3. Refactoring
4. Managing Breaking Changes
5. Syntax

## Steps To Migrate AngularJS to Angular

1. Assess Your AngularJS Application
2. Update Dependencies
3. Establish A New Angular Application
4. Transform Controllers into Components
5. The Angular CLI Commands
6. The Angular Upgrade Module
7. Validate Your New Angular Application

## Conclusion

## FAQs

1. What are the benefits of migrating to Angular?
2. Does Angular offer browser compatibility?
3. What is the latest version of Angular and AngularJS?

Hire Best Web Application Development Company

# Introduction

Angular has always been in the talks since its launch in 2016. Many business owners, whether new or old, preferred it for their web app development in 2024. Initially, with the AngularJS launch in 2010, the tech development marketplace accepted it with open hands. Business owners used it for its web app development, but with the official announcement of AngularJS being deprecated on December 31, 2021, many business owners shifted their focus to Angular. According to the [StackOverflow Developer Survey 2023](#), Angular is the fifth most popular technological stack among the development community worldwide, with 17.46% of votes. In contrast, AngularJS ranks sixteenth with only 7.21% of votes.

This makes migrating AngularJS to Angular even more crucial for business owners to keep up with the evolving market trends. However, before we proceed with the details of migrations and other aspects, let us first glimpse the comparison between Angular and AngularJS.

# AngularJS and Angular – Comparison Table

Below are a few points relating to the circumferential comparison between the two.

Parameter	AngularJS	Angular
Language	JavaScript	TypeScript
Architecture	MVC	Component-Based
Directives	Extensive Set of Directives	Standard Directives
Data Binding	Supports two-way data binding between view and model using {} or []	Uses ng-bind for one-way data binding and ng-model for two-way data binding
Scoping	Uses Scopes	No Scopes
Development Tools	Limited	In-built CLI
Mobile Support	No direct support	Supports mobile development with frameworks like Ionic and NativeScript
Current Status	Limited support ended on December 2021	Actively maintained and updated
Material Design	Need third-party library support	Built-in support
Performance	Limited performance due to two-way binding and DOM	High performance due to continuous updates and no binding

## Why Migrate AngularJS to Angular?

The main reason for upgrading AngularJS to Angular is that Angular is the later version and is actively updated and maintained by the official team. In contrast, AngularJS is the deprecated version that is no longer maintained. The other such reasons for migrating AngularJS to Angular are:

## Community and Ecosystem

Having been backed by Google, Angular has a vibrant and active community with a wealth of third-party libraries, tools, and resources available for developers. AngularJS, on the other hand, lacks extensive community support. Migrating AngularJS to Angular allows you to gain access to this extensive community and ecosystem along with tutorials, libraries, and more, which, in the long run, simplifies finding solutions to common problems and keeps you updated with the best practices.

## Timely Updates

The core team at Angular (actively maintained by Google) promptly releases new framework versions, with the latest version of **Angular v17** released recently. Angular offers long-term versions to ensure your app remains supported and receives timely security updates. The last update of AngularJS was in December 2021, after which it was deprecated.

These updates are vital in keeping the framework robust, secure, and more productive for your business application and its end-users. You get significant protection against data breaches and security vulnerabilities.

## Third-Party Compatibility

The AngularJS to Angular migration allows for increased support for performance and compatibility with extensive third-party tools and libraries available at your disposal. The upgrade offers you the ability to create large-scale enterprise applications effectively.

AngularJS also offered flexibility for creating large-scale applications but lacked stability. However, with its component-based approach, Angular offers particular methods to exchange among components. It also offers a comprehensive ecosystem for your development teams and simplified Integrated Development Environment (IDE) integrations for improved efficiency, making it easier for your development team to build applications.

## **Confused! If Migrating AngularJS To Angular Is The Right Choice For You?**

Hire JavaScript Developers from us to gain insight and realize the true benefits of upgrading AngularJS to Angular for your business app.

**Hire Now**

## **Improved Performance**

Angular is ideally designed with performance in mind, leveraging features such as ahead-of-time compilation (allows for faster app loads) and improved change detection mechanisms, resulting in faster rendering and better overall performance than AngularJS. When migrating from AngularJS to Angular, you easily achieve scalability and flexibility, reducing your application's overall bundle size.

However, with AngularJS, the performance is always a third-party dependency metric, which is complex to handle later. Angular also employs a command-line interface that reduces development time and simplifies development.

## **Setting Up the AngularJS to Angular Migration**

You can migrate your existing AngularJS application to the Angular framework in record time by following the steps below, depending on the complexity of your application and the anticipated timeline, to ensure a smooth transition.

## Plan Your Migration Strategy

When initiating your migration process, effectively plan your project's process, resources, technologies, and timelines. Initially, you must also determine the components of modules that you wish to implement first, migrate to TypeScript, and implement Angular component directives for the development process.

You must also identify the non-essential features and functionalities within your AngularJS applications, including outdated and less-used widgets. This will help you strategize the critical components and features during migration, minimizing disruption.

Set up the **TypeScript compiler** before starting the migration process. You can use the **@types/angular package** and **TypeScript module integration** to interoperate with Angular's JavaScript.

## Create a New Build

To migrate your business application, you must create a new build for your legacy applications using the Angular CLI or Webpack. Copy your AngularJS files to a new folder in the Angular structure; it will help you establish a connection that automatically retains the features of your application.

## Use the ngUpgrade Toolkit Library

Use this **ngUpgrade toolkit library** to mix and match the AngularJS and Angular components. The library here will bridge the two frameworks, enabling the Angular features to be integrated into the existing codebase without completely rewriting it.

Using this library helps you to migrate your software application in stages one step at a time, thereby reducing errors and unnecessary issues. It also allows you to test the

new functionalities and features before fully migrating from AngularJS.

## Transfer Services and Components

As the migration continues, you must downgrade the component features to make the Angular components available within the AngularJS part of your application. Also, you must correctly rewrite the specific sections of your code to ensure compatibility and seamless migration to the latest Angular version.

## Use a UI-router

Routing is a significant part of your migration process. When using the ***ngUpgrade***, the routing process is effectively and extensively supported by default, resulting in an easy transition. You can also use the UI router to ensure efficient navigation and migration of your application routes.

## The Ideal Path For Migration

You can choose either of the three paths when moving from AngularJS to Angular.

### Rewrite The Whole Application

As the name suggests, your development team will be responsible for rewriting the entire code, especially for outdated or legacy applications, which is a great option. Starting from the beginning, you can quickly re-design and rethink your application and implement new features as needed.

However, this approach faces a hiccup in that rewriting the entire application is a time-consuming process, meaning your development team would be required to work on both applications simultaneously, at least for some time, creating the new Angular applications and simultaneously supporting and maintaining the existing one for the current users.

### Hybrid with ngUpgrade



This allows you to keep running both frameworks simultaneously. Depending on your goals and priorities, you can achieve a slow and smooth transition that can last years or be unexpectedly quick. Your development team can choose one route at a time and start upgrading from the bottom up. The ngUpgrade allows you to use Angular and AngularJS to exist simultaneously and work together without disrupting the application.

## Angular Elements

It is a new package within the Angular ecosystem that enables you to transform the Angular components into custom elements natively understood by the browsers. The Angular Elements eliminate the need for ngUpgrade; however, they are helpful only for big and complex applications.

## Tools You Can Use To Migrate From AngularJS to Angular

As discussed, migrating to the Angular version is a manual and gradual approach. However, you can leverage the tools stated hereby to assist you with your migration process.

## NgUpgrade

Angular offers the ngUpgrade library that helps both AngularJS and Angular coexist simultaneously, allowing you to upgrade your application incrementally, one module or component at a time.

## Angular Upgrade Guide

This official [Angular documentation](#) offers detailed and comprehensive guidance on upgrading your applications to Angular. It also entails a step-by-step plan and recommendations that help handle various migration aspects.

# Angular CLI

This command line interface offers tools for generating Angular components, services, modules, and more. While it does not directly convert your AngularJS Code, it speeds up the development process by creating an equivalent Angular code.

## TSLint and Codelyzer

The TSLint is a static analysis tool for TypeScript and Codelyzer is a set of TSLint rules for Angular TypeScript projects. It helps identify the areas of Angular JS code that require your attention during the migration process.

## Manual Conversion Tools

Many online tools and scripts assist in manually converting specific AngularJS builds to Angular. However, these tools often target common patterns and are beneficial as starting points but inefficiently address all the scenarios.

## AngularJS to Angular Challenges

Migrating your existing application from AngularJS to Angular is a great choice. However, it comes with its own set of challenges, as discussed below:

### Project Scale and Compatibility

AngularJS is also limited when it comes to userbase support, which mandates an expansion of your project's scope during the update. Also, it verifies compatibility with third-party libraries, which is a point to keep into consideration.

### Command Line Interface

Angular has its own CLI, which requires you to adjust your project structure development workflow and the build processes accordingly to function correctly.

# Refactoring

Migrating the legacy JavaScript framework will affect your code due to the language difference. This requires your development team to rewrite sections of codes or even entire codes from scratch to adjust them appropriately for the upgrade.

## Managing Breaking Changes

The AngularJS to Angular upgrade poses a significant challenge in handling and addressing the breaking changes in the codebase. This requires you to stay updated with the latest version alterations and thorough testing to ensure quality and efficiency throughout the migration process.

## Syntax

Adapting ideally to the new syntax conventions, directives, and data components is a challenge within the migration process. Keeping yourself updated with the TypeScript syntax and its role within the Angular architecture can significantly help you for a successful upgrade.

## Steps To Migrate AngularJS to Angular

Migrating from AngularJS to Angular sometimes seems a tough nut to crack even to experienced developers. You can follow the steps below and easily upgrade to Angular without any hurdles.

### Assess Your AngularJS Application

Before you move ahead with the migration process, you must assess your current AngularJS application and understand its structure and functionalities. This will help you find the areas which require special attention.

### Update Dependencies

Then, update all the dependencies, including the AngularJS, to the latest version before you move ahead with the migration process. This will prevent any future hiccups during the migration process.

## **Establish A New Angular Application**

Set up the new Angular application, and then initiate the transfer of your existing AngularJS codebase into the new Angular environment.

## **Transform Controllers into Components**

Easily transition your AngularJS controllers into the Angular components, aligning with your Angular's component-based architecture.

## **The Angular CLI Commands**

Use the Angular CLI commands to streamline the code generation process and the application execution process efficiently.

## **The Angular Upgrade Module**

Leverage the Angular upgrade module to integrate the AngularJS and Angular code within the same application environment.

## **Validate Your New Angular Application**

Perform comprehensive testing of your newly migrated Angular application to verify that all the functionalities and features work as intended.

## **Conclusion**

As the official support for the AngularJS version has been deprecated, it is a great choice to upgrade AngularJS to Angular. This will boost your development approach and keep your business app secure and performant. However, migrating to Angular has its hiccups and pros and cons.

It is important to evaluate and assess your project goals and requirements and make an informed choice on how you should upgrade your AngularJS application to the Angular framework. If you find it complex or are facing challenges, you can also get in touch with an [Angular Development Company](#) like Positiwise and utilize their expertise in Angular Migration Service to make your business application stand out among the competitors.

## FAQs

### What are the benefits of migrating to Angular?

The benefits of migrating to Angular include improved performance, better developer tools, and long-term support for your business applications.

### Does Angular offer browser compatibility?

Yes, Angular does offer browser compatibility due to its modern features and optimized code that you can leverage in your business app development.

### What is the latest version of Angular and AngularJS?

The last version of Angular is v17, recently released on 8 November 2023 with exceptional new features and updates. However, the last version of AngularJS was v1.8, released in 2016.



**Jemin Desai** 

**Verified** Expert in Marketing Strategy and Brand Recognition

Jemin Desai is [Chief Marketing Officer](#) at Positiwise Software Pvt Ltd, he is responsible for creating and accelerating the company's marketing strategy and brand recognition across the globe. He has more than 20 years of experience in senior marketing roles at the Inc. 5000 Fastest-Growing Private Companies.

## Hire Best Web Application Development Company

STARTS 15 DAYS RISK-FREE TRIAL

### Related Posts



What is New in Angular 15? How to Upgrade to Angular 15?



What's New in Angular 17: Exploring the Latest Features

## SERVICES

Custom Software Development

Web App Development

eCommerce Development

Mobile App Development

Software Testing & QA

UI UX Design Services

## TECHNOLOGIES

.NET

PHP

WordPress

Full Stack .NET

Magento

PPC Services

## HIRE A TEAM

Hire .NET Developers

Hire Python Developers

Hire PHP Developers

Hire Full Stack Developers

Hire Backend Developers

Hire Front-End Developers



# CONTACT INFO

+91 78020 28008



## INDIA OFFICE

B 712-714, Shilp Corporate Park, Rajpath Rangoli Road,Bodakdev,  
Ahmedabad – 380054



## USA OFFICE

1309 Coffeen Avenue STE 1200, Sheridan, Wyoming, 82801, USA

© Copyright 2024 | Positiwise Software Private Limited | All Rights Reserved

[Privacy Policy](#)   [Refund Policy](#)   [Terms of Use](#)   [Sitemap](#)