

# COMPARATIVE ANALYSIS OF ANGULAR VERSIONS TECHNOLOGY

---

Ms. Sayali M. Ikale<sup>1</sup>

Tester, LiberPays, Mumbai, India, [sayali.ikale@liberpays.com](mailto:sayali.ikale@liberpays.com)

---

## ABSTRACT:

In recent years, there are many angular versions like angularjs, angular2, angular3, angular4, angular5, angular6, angular7, etc. For developers to choose this research paper include how angularjs changes features from angularjs to angular 7 and what features they included in current version. When angularjs first comes in technology use then that time we use some links and files for angularjs for adding features in our project but now days we used readymade set up and front end for single page web development of websites. In this research paper we are also going to discuss advantages & disadvantages, features regarding binding elements, modules, filters, directives. Attempting to interpret that, the changes and upgrades Angular has gone through to offer better value to its users can be best depicted by comparing different versions and updates it has come up with in the form of AngularJS, Angular 2, Angular 4 and Angular 5.

**Keywords:** Angularjs, Angular 2, Angular 3, Angular 5, Angular 6, Angular 7

## 1. INTRODUCTION

The developers should know how to choose the most suitable versions with the best support features to use in their project. This research paper helps to developer to choose version of angularjs & discuss the various versions of angularjs that are used in a single page front-end development of websites.

AngularJS was created by Misko Hevery.angularjs is basically used for single-page web application development. In angularjs we used typescript and JavaScript for dynamic part of application. Angularjs is maintained by Google [1].Angularjs is a nothing but open-source software. Angularjs is a structural framework for dynamic web application [2].MVC Means

Model view controller is a software architecture pattern that separates representation from user interaction. In angularjs we used MVC framework. Generally, the model consists of application data and functions that interact with it, while the view presents this data to the user; the controller mediates between the two [2].

It is nothing but a client side technology .In these we used JavaScript for client side scripting and typescript for server as well as client side scripting. It works with the long-established technologies of the web Like html, css and JavaScript to make the One page application web development easier and faster [2].designers can use HTML as the template language. We can also use angular to build any application using advantages of features like two-way binding, material, decorators, dependency injection, templating, restful api handling, Ajax handling, modularization[3].

For angular 2 we have use some commands for installing angular using command prompt in angular 2 they give readymade user interface for designing application. There are separate coding and static part are different. So we use ng model for initializing application.

In angular 3 there are some issue occurred so because of that angular 3 is cancelled and angular 4 is released and In angular 4 they include new features like ng-template and if and if else, two way binding. Because angular2 provides only one way binding features.

Angular 5 doesn't bring any significant change from Angular 4. However, lots of new features and many new improvements are done in this version for eg. New Router Lifecycle Event, Compiler Improvements, Optimization with HttpClient Feature, Internationalized Date & Currency.

New features added to Angular 6 like Updated Angular CLI, Command Line interface, Updated CDK, Component Development Kit, Updated Angular Material, Usage of RxJS, a reactive JS library, Angular Element.

Again new features added to Angular 7, CLI Prompts, Application performance, Angular Material & the CDK, Virtual Scrolling, Drag and Drop, Angular Compatibility Compiler (ngcc), Angular Do-Bootstrap, Better Error Handling [8].

Comparing between the angularjs and all versions to check which version is beneficial for developers and provide new features to developers to implement single page web development.

The difference is discussed in this paper. Comparing of all versions can be done using npm commands which can give the interface for application and show it in a browser. Differentiated & compared with the fact to determine the best way from the available versions.

## **2. LITERATURE VIEW**

Emma Thorén has presented a literature review on creation and purpose of angular. Usage of frameworks is a crucial technology to be successful in developing business, Angular has exploded in usage because of its unique features. To give an as accurate knowledge as possible this essay contains a literature study of reliable references. The essay also contains an empirical study based on developers own perspective of Angular. With the help of this essay you will have the opportunity to choose which version of Angular suits your company best [4].

Nilesh Jain has reviewed an article based on AngularJS: A Modern MVC Framework in JavaScript and discussed the various features of angularjs and also the comparative analysis of angularjs, backbone.js, and ember.js.they also give suggestion for why to choose angularjs [5].

Pankaj kumar has reviewed an article based on Understanding the Differences in versions of AngularJS vs. Angular 2 vs. Angular 4.it gives features, history, advantages. Keep reading to find out what has changed in Angular and why migrating to the latest version is a good idea.it gives an in-depth comparison so you can understand the differences and make an informed decision [6].

Miguel Ramos has presented a survey study on angularjs performance. .In this paper, it gives the report the results of a survey with 95 professional developers about performance issues of single page web applications.it gives reports i.e. followed by developers to avoid performance problems (e.g., use of third-party or custom components), the general causes of performance problems in single page web applications, and the technical and specific causes of performance problems [7].

Karsten Sitterberg and Thomas Kruse show all the innovations of angular 4 in the article and shared all top features of angular 4 that we need to know [9].Manish has reviewed an article. This article is a part of the series i.e. make a crud web app with angular 5,bootstrapping,inside components, creating new component, routing[10].

Ahmed Abdelsalamhas reviewed an article based on Angular 6 and the new amazing features. Each part in this article is not intended to be a complete guide, but rather an overview of the basics to get you Up and running so you can get to know how to collect things together and understand what the framework has to offer. [11].Todd Palmerhas reviewed an article based on angular 7 more. This article is gives a new stuff in Angular version 7[12].

### 3. PROCESS MODEL

There are several differences in angularjs and angular 2

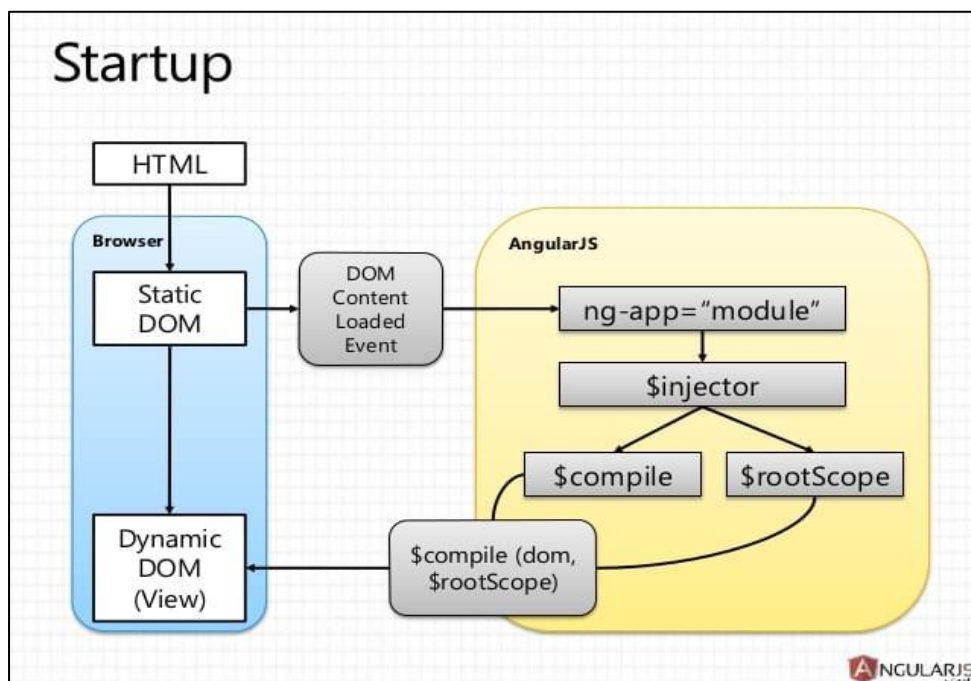
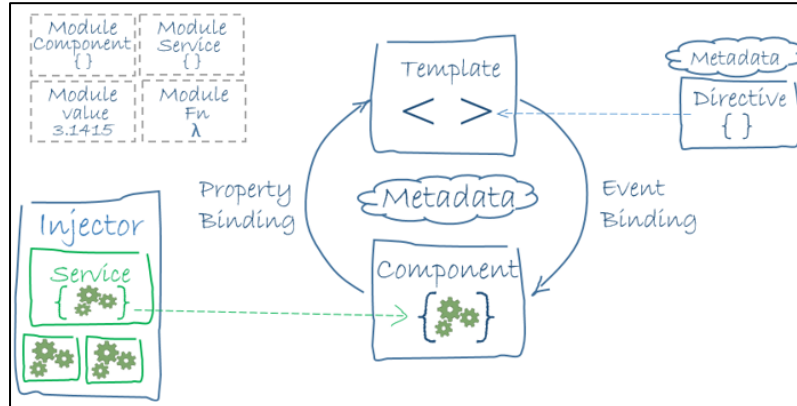


Fig. 1: Process of angular js using ng-app[13]

#### Angular 1 Controller:-

```
var app = angular.module("userApp", []);
app.controller("productController", function($scope) {
    $scope.users = [{ name: "Anil Singh", Age:30, department : "IT" },
    { name: "Aradhya Singh", Age:3, department : "MGMT" }]; }); [13]
```

## Angular 2 Components using TypeScript:-



**Fig.2:** Process of angular 2 using template and component [13]

## Templates in AngularJS and Angular 2

In AngularJS the template is written using HTML. For making it dynamic application, you can add AngularJS-specific code such as attributes, markups, filters and form controls. It also supports the two-way data binding technique. Following code shows how to use directives.

```
<html ng-app>
<!-- Body tag augmented with ngController directive -->
<body ng-controller="MyController">
<input#t ng-model="foo" value="bar">
<!-- Button tag with ngClick directive -->
<!-- Curly bracket is a template binding syntax -->
  button ng-click="changeFoo()">{{ buttonText }}</button>
<script src="angular.js"></script>
</body>
</html>[13]
```

In Angular, AngularJS template structure was reworked and lots of new features were added to the templates.

## **Dependency Injection in AngularJS and Angular 2**

Dependency Injection is a design pattern that takes care of satisfying dependencies and injecting them into the components when they're required. The factory method is registered with a module named myModule:

```
angular.module('myModule', [])  
.factory('serviceId', ['depService', function(depService) {  
  // ...  
}])  
.directive('directiveName', ['depService', function(depService) {  
  // ...  
}])  
.filter('filterName', ['depService', function(depService) {  
  // ...  
}]);[13]
```

This approach is Angular has a newer dependency injection system that's different from that of the older DI pattern. Angular's dependency injection is managed through the `@NgModule` array that comprises `providers` and `declarations`. The `declarations` array is the space where components and directives are declared.

## **Bootstrapping in Angular 1 using ng-app,**

```
angular.element(document).ready(function() {  
  angular.bootstrap(document, ['userApp']);  
});[13]
```

## **Bootstrapping in Angular 2,**

```
import { bootstrap } from 'angular2/platform/browser';  
import { Users Component } from './product.component';  
bootstrap(UserComponent);[6]
```

In Angular js is upgraded to Angular2 structural directives syntax is changed to **ng-repeat** is replaced with **\*ngFor** etc.

For example as,

//Angular 1,

```
<div ng-repeat="user in users">
```

```
  Name: {{user.name}}, Age : {{user.Age}}, Dept: {{user.Department}}
```

```
</div>[6]
```

//Angular2,

```
<div *ngFor="let user of users">
```

```
  Name: {{user.name}}, Age : {{user.Age}}, Dept: {{user.Department}}</div>[6]
```

There are several differences in angular 4 and angular 2

The screenshot shows two side-by-side code editors. The left editor, titled 'ng-2.html', shows the Angular 2 syntax for an email input field with a complex regular expression pattern. The right editor, titled 'ng-4.html', shows the Angular 4 syntax for the same input field, which is significantly simpler and more concise.

```

ng-2.html
1  Angular2
2  <input
3  type="email"
4  name="email"
5  required
6  pattern=
7    "[a-z0-9!#$%&'*/+\\/=/?^_
8    `{|}~.-]+@[a-z0-9]([a-z0-9-]*[a-z0-9])?
9    (\\.[a-z0-9]([a-z0-9-]*[a-z0-9])?)"
10 />
11

ng-4.html
1  Angular4
2  <input
3  type="email"
4  name="email"
5  required
6  email
7  />
8

```

Fig.3: Differences between the syntax of angular 4 and angular 2[6]

The screenshot shows two side-by-side code editors. The left editor, titled 'ng-2.html', shows the Angular 2 syntax for conditional rendering using \*ngIf. The right editor, titled 'ng-4.html', shows the Angular 4 syntax, which uses \*ngIf for the primary condition and an #elseBlock directive for the alternative content.

```

ng-2.html
1  <-- Angular2 -->
2  <p *ngIf="condition">
3  Some Stuff
4  </p>
5  <p *ngIf="!condition">
6  Some Other Stuff
7  </p>
8

ng-4.html
1  <-- Angular4 -->
2  <p *ngIf="condition ; else elseBlock">
3  Some Stuff
4  </p>
5  <ng-template #elseBlock>
6  <p>
7  Some Other Stuff
8  </p>
9  </ng-template>
10

```

Fig.4: Differences between the syntax of angular 4 and angular 2[6]

### **Disadvantages of Angular 3**

Because of misalignment of router package in @angular/router was using v3.3 already, so they move to Angular 4 Finally.

#### **Test:**

/Angular 4 -

```
 TestBed.overrideTemplate(UsersComponent, '<h2>{{ users.name }}</h2>');[6]
```

//Angular 2 -

```
 TestBed.overrideComponent(UsersComponent, {
  set: { template: '<h2>{{ users.name }}</h2>' }
});[6]
```

#### **Http:**

//Angular 4 -

```
 http.get(`${baseUrl}/api/users`, { params: { sort: 'ascending' } });[6]
```

//Angular 2-

```
 const params = new URLSearchParams();
 params.append('sort', 'ascending');
 http.get(`${baseUrl}/api/users`, { search: params });[6]
```

### **There are several differences in Angular 6 vs Angular 5 vs Angular 4**

#### **RxJS 6 Related import paths -**

In angular 5 we used ,[6]

```
 import { Observable } from 'rxjs/Observable';
 import { Subject } from 'rxjs/Subject';
```

In angular 6,Use a single import -[6]

```
 import { Observable, Subject } from 'rxjs';
```



### **The <template> deprecated, Now Angular 6 introduce <ng-template> –**

Now in Angular 6, you should use <ng-template> instead of <template>

For example, previously you are using[6]

```
<template [ngIf]="IsAdmin">
  <p>This template renders only if IsAdmin is true.</p>
</template>
```

Now in Angular 6, you should use <ng-template> instead of <template>

```
<ng-template [ngIf]="IsAdmin">
  <p>This template renders only if IsAdmin is true.</p>
</ng-template>
```

### **Service level changes (the way of marking a service as global)**

In the previous versions, if we want to gives a service to the full single page application –we should add it to *providers []* in the AppModule but in the Angular 6 released we should not add in the *providers []* in the AppModule.

Example for marking a service as global -

Instead of

```
//my.service.ts
export class MyService { }

//In app.module.ts
//JavaScript imports services
import { MyService } from './my-serice.service';
//AppModule class with the @NgModule decorator
```

```
@NgModule({
  declarations: [],
  providers: [MyService]
})
```

```
export class AppModule {  
  //exporting app module  
}
```

Use with Angular 6 released-

```
//my.service.ts  
@Injectable({providedIn: 'root'})  
export class MyService { }  
@NgModule({  
  declarations: [],  
  providers: []  
})
```

```
export class AppModule { }
```

NgModelChange:

Previously -

```
<input [(ngModel)]="name" (ngModelChange)="onChange($event)">
```

And

```
onChange(value) {  
  console.log(value); // would log the updated value, not old value  
}
```

Now Use -

```
<input #modelDir="ngModel" [(ngModel)]="name"  
(ngModelChange)="onChange(modelDir)">
```

And

```
onChange(NgModel: NgModel) {  
  console.log(NgModel.value); // would log old value, not updated value  
}[6]
```

Angular 7 focused more on upgrading the current features and rather than publishing as much features as Angular 6, Angular 7 added a new compiler that is called Angular Compatibility Compiler (ngcc).

## **DoBootstrap-**

Angular 7 added a new lifecycle hook that is called ngDoBootstrap and an interface that is called DoBootstrap.

The following Example -

```
//ngDoBootstrap - Life-Cycle Hook Interface
class AppModule implements DoBootstrap {
    ngDoBootstrap(appRef: ApplicationRef) {
        appRef.bootstrap(AppComponent);
    }
}[12]
```

## **5. CONCLUSION**

In the last few years the angular technology is increased rapidly. From the above study of the differences and history, features of the angular versions. Angularjs is still useful or else everyone would have switched to Angular 2 or Angular 7 by now, and many mores comes with lots of new features.

Each version of angular meaningful benefits, but there is much to gain in being up-to-date with the latest version. Angular is decidedly speedy than Angularjs, has a mobile-driven approach, executes better with components, and allows smoother relocation from previous versions. If it is comfortable with one version of angular, migrating to another should easy and well worth the effort.

## **6. FUTURE ENHANCEMENT**

In future, angular 7 is switched to Angular 8.0, a upgrade to Google's popular framework now in beta release, will feature Ivy, a new renderer. Google also has change additional features planned for Angular 8.0, including better loading for JavaScript code.

Angular provides dependency injection, which is particularly useful for assembling data services for applications, along with use of HTML template to create components. In Angular, developers

still make components with an HTML component that attached to Typescript code for essential parts of the program.

## **REFERENCES**

- [1] *AngularJS is a structural framework for dynamic web ... International Journal of Scientific&Engineering Research, Volume 7, Issue 2, February-2016*
- [2] *ng-book. The Complete Guide to Angularjs. Written by ©2013 Ari Lerner ISBN 978-0-9913446-0-4*
- [3] *ng-book: The Complete Guide to Angular4 Written by Nate Murray, Felipe Coury*
- [4] Usage of Angular from developer's perspective Based on a literature and empirical study By Emma Thorén Filip Brännlund Stål
- [5] AngularJS: A Modern MVC Framework in JavaScript Nilesh Jain et al, Journal of Global Research in Computer Science, 5 (12), December 2014,17-23
- [6] AngularJS Vs. Angular 2 Vs. Angular 4: Understanding the Differences By Pankaj Kumar
- [7] AngularJS Performance: A Survey Study Miguel Ramos<sup>1</sup>, Marco Tulio Valente<sup>1</sup>, Ricardo Terra<sup>2</sup>
- [8] Top 10 Features Of Angular 7.0 article by angularminds
- [9] angular-4-top-features article by Karsten Sitterberg, Thomas Kruse, april 13,2017
- [10] Angular 5—routing article-series by manish, Feb 16,2018
- [11] angular-6-article-series by ahmed abdel salam , may 5,2018
- [12] Angular 7 in the House! A quick look at some of the new stuff in Angular version 7 by todd palmer nov 1,2018
- [13] AngularJS and Angular 2+: a Detailed Comparison By Manjunath M, April 06, 2018