**Angular JS**

Angular is a JavaScript framework which allows you to create reactive SPAs (Single Page Applications) **Single**-**Page Applications** (SPAs) are Web **apps** that load a **single** HTML **page** and dynamically update that **page** as the user interacts with the **app**.

Angular JS is a popular **JavaScript Frontend** Framework for creating web application which was released by **Google** in the year 2010. It became quite popular because of its rich features and built-in functionality to create reactive and responsive web apps.

**Angular Versions & Release Dates**

**Initial release October 20, 2010 developed by Google**

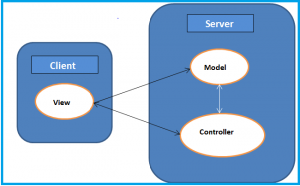
* Version 2: Sept 2016
* Version 4: March 2017
* Version 5: Oct 2017
* Version 6: March 2018
* Version 7: Oct 2018

Angular 2 moved to Beta in December 2015, and the first release candidate was published in **May 2016**. The final version was released on **September 14, 2016**.

The plan is to release a major version every six months, so Angular 5 will show up in **September 2017**, followed by Angular 6 in **March 2018** and Angular 7 in **September 2018**. Nobody can predict the future, but the idea is for future versions to introduce minimal breaking changes.

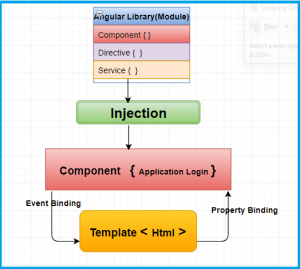
**Architecture**

**Angular 1**



* The architecture of Angular 1 is based on the **Model View Controller**.
* Angular 1 application was built using JavaScript.

**Angular 2**



* Angular 2 is based on a **Components** structure, like what we see in React.js
* Angular 2 applications were built on **Typescript**,which is a superset of JavaScript
* Angular 2 was focused on **mobile apps development**

**Angular 4**

Angular 4 is much faster and reduces the file generated code of components; it also allows the developer to generate code you can use in debug mode and production mode.

**Angular 1.x vs. Angular 2**

Angular v1.0 use JavaScript to build the application while Angular v2.0 uses the Typescript to write the application.

TypeScript is a superset of JavaScript which helps to build more robust and structured code.

Angular 2 uses **TypeScript 2.0**

Angular v2.0 has changed to component based UI. This helps a developer to divide the applications in terms of components with desired features and enable to call required UI.

These have helped to improve the flexibility and reusability as compared to Angular v1.0

* The controller concept which was present in Angular v1.0 is eliminated in Angular v2.0.
* It is entirely component based.
* Angular2 has better performance.
* Angular2 much more testable
* Angular2 provides to nested level components.
* Ahead of Time compilation (AOT) improves rendering speed
* Angular2 execute run more than two programs at the same time.
* Angular1 is controllers and $scope based but Angular2 is component based
* The Angular2 structural directives syntax is changed like ng-repeat is replaced with \*ngFor etc.
* Angular 1.x controllers and $scope are gone.
* Angular 2 uses camel Case syntax for built-in directives.
* Two-way data binding: ng-model replaced with [(ngModel)]
* Way of Bootstrapping Angular Application is changed
* Angular 2 is 5 times faster as compared to Angular 1

**Angular 2 vs. Angular 4**

* Angular v4.0 is compatible with newer versions **TypeScript 2.1** and **TypeScript 2.2.** This helps with better type checking and also enhanced IDE features for Visual Studio Code
* The view engine introduced Angular 4, which decreased the size of the generated code using the [**Ahead of Time**](https://angular.io/guide/aot-compiler) (AOT) manner. It provides great impact on application performance and security.

**Features of Angular version 4.0**

Angular 4 contains some additional Enhancement and Improvement. Consider the following enhancements.

**1. Smaller & Faster Apps** - Angular 4 applications is smaller & faster in comparison with Angular 2.

**2. View Engine Size Reduce**- Some changes under to hood to what **AOT**generated code compilation that means in Angular 4, improved the compilation time. These changes reduce around 60% size in most cases. The bundles are reduced to thousands of KBs.

**3. Animation Package**- Animations now have their own package i.e. @angular/platform-browser/animations

**4. Improvement** - Some Improvement on \***ngIf**and \***ngFor**. Angular 4 this was solved with the help of ‘else’.

**5. Template** - The template is now **ng-template**. You should use the “ng-template” tag instead of “**template**”. Now Angular has its own template tag that is called “ng-template”.

**6. NgIf with Else** – Now in Angular 4, possible to use an else syntax as,

<div \*ngIf="user.length > 0; else empty"><h2>Users</h2></div>

<ng-template #empty><h2>No users.</h2></ng-template>

**7. AS keyword** – A new addition to the template syntax is the “**as**keyword” is use to simplify to the “**let**” syntax.

Use of as keyword,

<div \*ngFor="let user of users | slice:0:2 as total; index as = i">

{{i+1}}/{{total.length}}: {{user.name}}

</div>

To subscribe only once to a pipe “|” with “**async**” and If a user is an observable, you can now use to write,

<div \*ngIf="users | async as usersModel">

<h2>{{ usersModel.name }}</h2> <small>{{ usersModel.age }}</small>

</div>

**8. Pipes** - Angular 4 introduced a new “**titlecase**” pipe “|” and use to changes the first letter of each word into the uppercase.   
  
The example as,

<h2>{{ 'anil singh' | titlecase }}</h2>

<!-- OUPPUT - It will display 'Anil Singh' -->

**9. Http** - Adding search parameters to an “**HTTP**request” has been simplified as,

//Angular 4 -

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

//Angular 2-

**const** params = **new** URLSearchParams();

params.append('sort', 'ascending');

http.get(`${baseUrl}/api/users`, { search: **params** });

**10. Test**- Angular 4, overriding a template in a test has also been simplified as,

//Angular 4 -

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

//Angular 2 -

TestBed.overrideComponent(UsersComponent, {

set: { template: '<h2>{{users.name}}</h2>' }

});

**11. Service**- A new service has been introduced to easily get or update “**Meta Tags**” i.e.

@Component({

selector: 'users-app',

template: `<h1>Users</h1>`

})

**export** **class** UsersAppComponent {

**constructor**(meta: **Meta**) {

meta.addTag({ name: 'Blogger', content: 'Anil Singh' });

}

}

**12. Forms Validators** - One new validator joins the existing “required”, “minLength”, “maxLength” and “pattern”. An email helps you validate that the input is a valid email.

**13. Compare Select Options** - A new “**compareWith**” directive has been added and it used to help you compare options from a select.

<select [compareWith]="byUId" [(ngModel)]="selectedUsers">

<option \*ngFor="let user of users" [ngValue]="user.UId">{{user.name}}</option>

</select>

**14. CanDeactivate** - This “**CanDeactivate**” interface now has an extra (optional) parameter and it is containing the next state.

**Features of Angular version 4.0**

Angular 5 is going to be a much better Angular and you will be able to take advantage of it much easier. 

The Angular 5 contains bunch of new features, performance improvements and lot of bug fixes and also some surprises to Angular lovers.

1)    Make AOT the default - Ahead-of-Time Compiler

2)    Watch mode

3)    Type checking in templates

4)    More flexible metadata

5)    Remove \*.ngfactory.ts files

6)    Better error messages

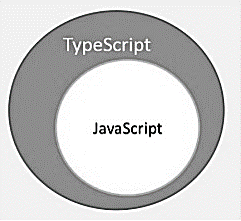
7)    Smooth upgrades

8)    Tree-Shakeable components

9)    Hybrid Upgrade Application

# Type Script

TypeScript is a strongly-typed superset of JavaScript, which means it adds some syntactical benefits to the language while still letting you write normal JavaScript if you want to. It encourages software developers to more declarative style of programming like interfaces and static typing, offers modules and classes, and most importantly, integrates relatively well with popular JavaScript libraries and code. It totally follows the OOPS concept.



Or we can say TypeScript is a transpiler.

**Transpiler**

It converts one language into another language.  
  
**How to install it**

There are two main ways to install the TypeScript tools:

* Via npm (Node.js Package Manager)
* By installing, TypeScript via Visual Studio