

WHAT IS ANGULAR 4?

- AngularJS used to be the “golden child” among JavaScript frameworks, as it was initially introduced by Google corporation in 2012. It was built with the Model-View-Controller concept in mind, though authors of the framework often called it “Model-View-*” or even “Model-View-Whatever”.
- It is a TypeScript-based open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations.
- Angular 4 released in March 2017 proves to be a major breakthrough and is the latest release from the Angular team after Angular2.
- Angular 4 is almost the same as Angular 2. It has a backward compatibility with Angular 2. Projects developed in Angular 2 will work without any issues with Angular 4.

WHAT IS ANGULAR 4?

- Angular 4 applications are smaller and faster when compared to Angular2. It uses the TypeScript version 2.2, the latest version(as of Angular4 release) which makes the final compilation small in size.
- Using angular we can build SPA(Single Page Applications)
- A single page application is super-simple to deploy if compared to more traditional server-side rendered applications: it's really just one index.html file, with a CSS bundle and a Javascript bundle.
- Another quite important thing was Dependency Injection, which allowed application components to be wired together in a way that facilitated reusable and testable code.

ADVANTAGES OF ANGULAR

- **Angular presents you not only the tools but also design patterns to build your project in a maintainable way.** When an Angular application is crafted properly, you don't end up with a tangle of classes and methods that are hard to modify and even harder to test. The code is structured conveniently and you won't need to spend much time in order to understand what is going on.
- **It's JavaScript, but better.** Angular is built with TypeScript, which in turn relies on JS ES6. You don't need to learn a totally new language, but you still receive features like static typing, interfaces, classes, namespaces, decorators etc.
- **No need to reinvent the bicycle.** With Angular, you already have lots of tools to start crafting the application right away. You have directives to give HTML elements dynamic behavior.

ADVANTAGES OF ANGULAR

- **Components are decoupled.** Angular strived to remove tight coupling between various components of the application. Injection happens in NodeJS-style and you may replace various components with ease.
- **All DOM manipulation happens where it should happen.** With Angular, you don't tightly couple presentation and the application's logic making your markup much cleaner and simpler
- **Testing is at the heart.** Angular is meant to be thoroughly tested and it supports both unit and end-to-end testing with tools like Jasmine and Protractor.
- **Angular is mobile and desktop-ready**, meaning you have one framework for multiple platforms.
- **Angular is actively maintained** and has a large community and ecosystem.

ADVANTAGES OF ANGULAR

- **Components are decoupled.** Angular strived to remove tight coupling between various components of the application. Injection happens in NodeJS-style and you may replace various components with ease.
- **All DOM manipulation happens where it should happen.** With Angular, you don't tightly couple presentation and the application's logic making your markup much cleaner and simpler
- **Testing is at the heart.** Angular is meant to be thoroughly tested and it supports both unit and end-to-end testing with tools like Jasmine and Protractor.
- **Angular is mobile and desktop-ready**, meaning you have one framework for multiple platforms.
- **Angular is actively maintained** and has a large community and ecosystem.

ENVIRONMENT SETUP

To install Angular 4, we require the following –

- Nodejs 6.9 or above
- Npm 3.x or above
- Angular CLI
- IDE for writing your code(preferably VSCode)

ENVIRONMENT SETUP

To install Angular 4, we require the following –

- Nodejs 6.9 or above
- Npm 3.x or above
- Angular CLI
- IDE for writing your code(preferably VSCode)

Download node.js from below link it will also install npm for you.

<https://nodejs.org/en/download/>

Download Vscode from below link

<https://code.visualstudio.com/download>

ENVIRONMENT SETUP

- Open node command prompt or the VSC and check for node and npm versions using
 - `node -v`
 - `npm -v`
- **@angular/cli** is an npm package and it's an amazing tool which helps with different repetitive operations when creating and working with Angular apps. You can install @angular/cli by typing the following command in your terminal.
 - `npm install @angular/cli -g`

ENVIRONMENT SETUP

- Once done with CLI installation try command “ng v ” to check the angular version and the dependencies installed with it in your machine.
- Create a new angular project using
ng new [projectname]
- Now get into your project folder by doing “cd projectname” and then run “ng serve” to run your angular app
- Angular runs on a default port number 4200 you can even run on your own customized port too by using below ng command
ng serve --host 0.0.0.0 --port 4205

ANGULAR APP PROJECT STRUCTURE AND FILES

The Angular 4 app folder has the following **folder structure** –

- **e2e** – end to end test folder. Mainly e2e is used for integration testing and helps ensure the application works fine.
- **node_modules** – The npm package installed is node_modules. You can open the folder and see the packages available.
- **src** – This folder is where we will work on the project using Angular 4.

ANGULAR APP PROJECT STRUCTURE AND FILES

The Angular 4 app folder has the following **files** –

- **.angular-cli.json** – It basically holds the project name, version of cli, etc.
- **.editorconfig** – This is the config file for the editor.
- **.gitignore** – A .gitignore file should be committed into the repository, in order to share the ignore rules with any other users that clone the repository.
- **karma.conf.js** – This is used for unit testing via the protractor. All the information required for the project is provided in karma.conf.js file.

ANGULAR APP PROJECT STRUCTURE AND FILES

- **package.json** – The package.json file tells which libraries will be installed into node_modules when you run npm install.
- **protractor.conf.js** – This is the testing configuration required for the application.
- **tsconfig.json** – This basically contains the compiler options required during compilation.
- **tslint.json** – This is the config file with rules to be considered while compiling.

ANGULAR APP PROJECT STRUCTURE AND FILES

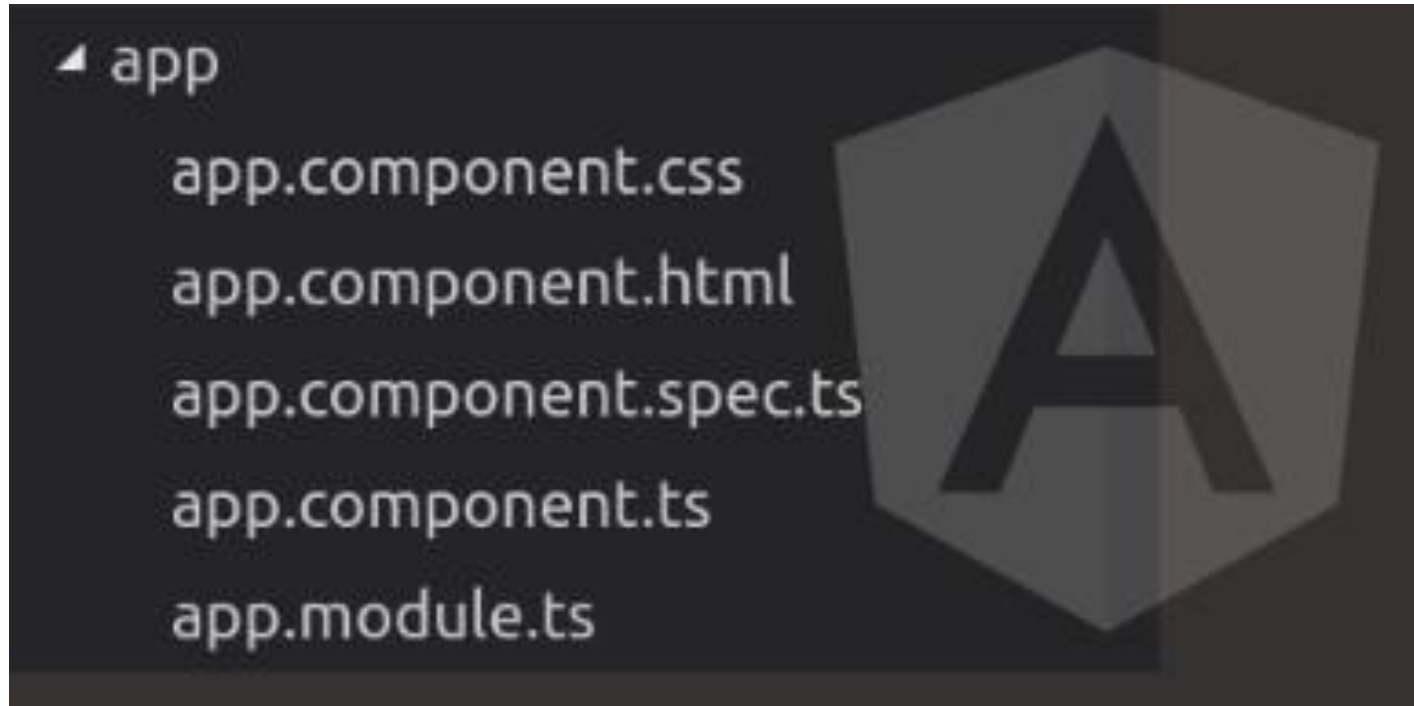
app :: app directory is where we define the building blocks of our angular project like modules , components , services etc.

assets :: as the name suggests , this directory contains all the static assets of our app like images etc.

environments :: environments directory hold our environment specific settings , e.g. different configuration files for development , testing and staging etc.

ANGULAR APP BUILDING BLOCKS

The app folder of your angular consists of



ANGULAR APP BUILDING BLOCKS

1. **app.module.ts** :: In this file , we declare our angular module , the @NgModule decorator is used to initialize different aspects of the angular app , observe that AppComponent is also declared.

```
// app.module.ts

import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

ANGULAR APP BUILDING BLOCKS

2. **app.components.ts** this file simply defines an angular component and this is where we have defined our app-root selector also.

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'app'; // this is used in the template
}
```


ANGULAR APP BUILDING BLOCKS

3. **app.component.html** this is the template file for our app component and this represents the visual part of our component which is rendered in the browser.

```
<div style="text-align:center">
  <h1>
    Welcome to {{title}}!! <!-- data inside curly braces comes from angular component -->
  </h1>
  
</div>
<h2>Here are some links to help you start: </h2>
<ul>
  <li>
    <h2><a target="_blank" href="https://angular.io/tutorial">Tour of Heroes</a></h2>
  </li>
  <li>
    <h2><a target="_blank" href="https://github.com/angular/angular-cli/wiki">CLI Documentation</a></h2>
  </li>
  <li>
    <h2><a target="_blank" href="http://angularjs.blogspot.ca/">Angular blog</a></h2>
  </li>
</ul>
```

ANGULAR APP BUILDING BLOCKS

4. **app.component.spec.ts** These are automatically generated files which contain unit tests for source component.

You can write unit testing to your angular app using frameworks like Karma, protractor and more.

Assets

You can save your images, js files in this folder.

Environment

This folder has the details for the production or the dev environment. The folder contains two files.

environment.prod.ts

environment.ts

ANGULAR APP BUILDING BLOCKS

More about app.module.ts: It has decorator @NgModule which has variables such as **declarations**, **imports**, **providers**, and **bootstrap**.

declarations – In declarations, the reference to the components is stored. The AppComponent is the default component that is created whenever a new project is initiated. We will learn about creating new components in a different section.

imports – This will have the modules imported as shown above. At present, BrowserModule is part of the imports which is imported from @angular/platform-browser.

providers – This will have reference to the services created. The service will be discussed in a subsequent chapter.

bootstrap – This has reference to the default component created, i.e., AppComponent.

ANGULAR APP BUILDING BLOCKS

main.ts

main.ts is the file from where we start our project development. It starts with importing the basic module which we need. Right now if you see angular/core, angular/platform-browser-dynamic, app.module and environment is imported by default during angular-cli installation and project setup.

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule);
```

As we can observe in the above snippet , the class AppModule is imported from the app.module.ts file in the app directory.

ANGULAR APP BUILDING BLOCKS

main.ts

main.ts is the file from where we start our project development. It starts with importing the basic module which we need. Right now if you see angular/core, angular/platform-browser-dynamic, app.module and environment is imported by default during angular-cli installation and project setup.

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule);
```

As we can observe in the above snippet , the class AppModule is imported from the app.module.ts file in the app directory.