# Angular

A framework for building client application in HTML,CSS and Javascript/TypeScript

## Structure

After running the command *ng new myApp* you will get a basic structure of files and folder like (I have taken only the important files in the view)

```
MYAPP
│   e2e
│   node_modules
│
└───src
│   └───App
│   |   |   app.component.css
│   |   |   app.component.html
│   |   |   app.component.spec.ts
│   |   |   app.component.ts
│   |   |   app.module.ts
│   |
│   |   Assets
│   |   environments
│   |   index.html
│   |   main.ts
│   |   polyfills.ts
│   |   styles.css
│   |   test.ts
│
│   editorconfig
│   angular.json
│   package.json
│   tsconfig.json
│   tslint.json
```

- **e2e** (*end to end*) : Used to write end to end test to simulate the real case scenario as an end user.
- **node_modules** : It contains all the third party libraries. This folder is only for development during deployment we bundle it first and deploy it. [TODO: Add more about the process of bundling]
- **src** : It contains actual source code.
  - App
  - Assets : Contains static assests used
  - environments : Information regarding the environment (like production, Qa)
  - **Main.ts** : It is starting point of our application. We do bootstraping of our main module in this file. [TODO: Add more about this process]

    ```
    platformBrowserDynamic().bootstrapModule(AppModule)
     .catch(err => console.log(err));
    ```

  - polyfills.ts : Angular uses features of JS that aren't available in current version of JS supported by most of the web browser out there, So this file fills the gap between features of JS that angular need and features that supported by current browser.

- .editorConfig : EditorConfig helps maintain consistent coding styles for multiple developers working on the same project across various editors and IDEs. All developers in project should use the same editorconfig.
- package.json : Contains version of our App and other dependencies.
  - dependencies : For running application in browser
  - devDependencies : For Development phase of application
- tsConfig.json : Settings for TypeScript Compiler. like target of JS version.
- tslint.json : Static Analysis tool for ts. It checks for
  - Readability error
  - Maintainability error
  - Functionality error

---

## Webpack

- It's a **[Build automation tool]** used by Angular-Cli.
- It does the following tasks for optimization :
  - It gets all our script and stylesheets and combine them
  - Put them in a bundle
  - Minifies that bundle
- When we execute *ng serve*, We get something like this

  ```
  chunk {main} main.js, main.js.map (main) 364 kB [entry] [rendered]
  chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 594 kB [entry] [rendered]
  chunk {runtime} runtime.js, runtime.js.map (runtime) 5.22 kB [entry] [rendered]
  chunk {styles} styles.js, styles.js.map (styles) 369 kB [entry] [rendered]
  chunk {vendor} vendor.js, vendor.js.map (vendor) 3.19 MB [initial] [rendered]
  ```

  What are these files?
  Bundles of polyfills, Stylesheet, Vendor etc

  - main.js : contains the code we written.
  - polyfills.js : It contains all the polyfills used in our App.
  - runtime.js :
  - styles.js : All stylesheets used in App. StyleSheets are actully stored in JS bundle. *vendor.js : All third party libraries.
- All the bundles generated by webpack are injected into our index.html as script in runtime.
  index.html we see in VsCode:

```html
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>MyApp</title>
<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
<app-root></app-root>
</body>
</html>
```

index.html we get in browser on viewing Page source

```html
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>MyApp</title>
<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
<app-root></app-root>
<script type="text/javascript" src="runtime.js"></script>
<script type="text/javascript" src="polyfills.js"></script>
<script type="text/javascript" src="styles.js"></script>
<script type="text/javascript" src="vendor.js"></script>
<script type="text/javascript" src="main.js"></script></body>
</html>
```

- **Hot Module Replacement/Reloading [HMR]** : It's a feature of webpack, Whenever a source file is edited, Webpack automatically refreshes the browser.

> **Note**: For better understanding of how angular works Kara Erickson: How Angular works.

---

## Angular Version History

Mainly there are two version of Angular

- Angular 1.x (Also called Angular JS)
- Angular 2.x+

> AngularJs (2010) ⇒ Angular 2.0 ⇒ Angular 2.1 ⇒ Angular 2.2 ⇒ Angular 2.3 ⇒ Angular 4.0 ⇒ ...
> Versioning why After 2.3 we got 4.0:

- Mainly Angular consists of these core libraries, this table has version of libraries during Angular 2.3.

| Library | Version |
|---|---|
| @angular/core | 2.3.0 |
| @angular/compiler | 2.3.0 |
| @angular/http | 2.3.0 |
| @angular/router | 3.3.0 |

These packages are distributed as seprate node packages via npm.
In order to align these versions they opted for angular 4.