Angular 2 :  
For single page application :

Angular2 cli .With cli it is easy. Need node js. Npm manager for various package manager.

1)Install angular-cli through the npm

npm install angular-cli –g

2)Create new project with name first-app inside the folder

ng new first-app

3)Start the development server :

ng serve

It also build our application.Angular 2 uses typescript.It converts the typescript back to the javascript. **Note** : Browser can not be able to run the typescript

Structure of the folder.

1. src/app folder contains all the ts files (Components).Type script
2. app.component.ts

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

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'app works!';

}

AppComponent is the typescript class.The @component decorator is attached to this class. The @Component decorator is imported from the angular2/core. It allows us to transform the normal typescript class into something else.Each component in our application need some html to display.

**Note** : stylesUrl is optional not the template.It is array so, we can use multiple files here for the css.

**Selector** : Allows us to use the component inside the html .Our custom html element. It decides how we want to insert the component to our html code.The **app-root** here refers to the html element app-root.It tells the angular 2 what to do when it enconter’s this  **app-root** inside the html **.**

**Different ways the selector are used :**

a)**#app-root**  - This is used whenever we are using the app-root as an id.

b)**.app-root** – This is used when we are using the app-root as an class of html element.ex : <span class=”app-root”></span>

Above we can see the decorator with @ sign.

Note : Angular 2 use the components to render our web application.each component not only contains the typescript but also have some html to display.

Template of AppComponent is stored in **app.component.html**

Css of AppComponent is stored in app.component.css.It is array type means it can contain more then one css.

**Now let’s see the html fir this file.**

<h1>

{{title}}

</h1>

{{}} is known as the string interpolation.It can display any property of our component dynamically.Here title is the property of the AppComponent class.

Inside the app folder we do have the spec file that is usually for the unit testing.

The different files are :

a)app.component.html – This is template for our component

b)app.component.css – This is css for out component

c)app.component..ts – This is out component file with ts extension that is typescript.

d)index.ts – This file keeps trak of all the files present inside the app fiolder. So, that the export from our folder can be easier.

**Environment Folder**

This folder contains some of the configurations which is needed for the compilation of our app. The default here is fine so, we not need to do any change here.Same for the other files. These are the files used by our cli(**Angular-cli**) to build our application.

The main files are :

a)**main.ts** : This is file that just starts our angular 2 application.

2)**styles.css**  - This file is used to apply the global css. That is applied to our whole application. While the styles applied to our app folder is applied directly to that component only.

3)**angular-cli.json**  - This files allow us to just change the setting of our angular-cli. For example where we want to store our source file etc.

4)**node\_modules** - contains all the dependencies of our project, We never change it . It is simply managed by our cli.

5)**dist folder**  - This is the folder where we never do any of the change. All the compiled codes are stored here by the cli. This we can deploy to the server for example.

**Note :** We mainly works inside the app folder.

**Typescript**

With creating the application with the help of angular-cli , the typescript get’s automatically installed but we can install it globally with the help of npm tool.

**npm install typescript –g**

**index.html –** Angular2 is a single page application.So it is not the component that is loaded but it is the index.html that is loaded.It the file that is loaded by our server.It is the single page application.

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>FirstApp</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>Loading...</app-root>

</body>

</html>

Above we can see **<app-root>Loading...</app-root>**.This is our component

**How the angular 2 app gets started ?**

Ans : There is no script we are seeing here.

Let us see our browser and see the source code there. We can see the 3 scripts download here.

a)inline.js

b)styles.bundle.js

c)main.bundle.js - **Imp**

These files are added during the compile time. **Main.bundle.js** contains all our code + angular2 code + all 3rd party package code. This also starts our angular application.

**Now, how it is started (main.ts)**

We can see the main.ts : It is the first file that get started by the bundle.It bootstrap our application.It starts our application by below line.

platformBrowserDynamic().bootstrapModule(**AppModule**);

**How the angular2 come to know which component to load ?**

Angular does not load the whole component. It just see it inside the AppModule.

Where the declaration of component is there. Just see the app.module.ts.

],

providers: [],

**bootstrap: [AppComponent]**

It bootstrap the AppComponent here.This tells the angular 2 that this component is the root component of the application. This is also present inside the index.html

**Understanding the AppModule Component**

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

import { NgModule } from '@angular/core';

import { FormsModule } from '@angular/forms';

import { HttpModule } from '@angular/http';

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

@NgModule({

declarations: [ **//Tells which directive we use in our application. Directives are the instructions that tells angular2 what to do.**

AppComponent

],

imports: [ /**/Tells which other modules do I use in our application.**

BrowserModule, **//These are predefined modules.**

FormsModule, **//when working with input fields**

HttpModule **//When working with http access**

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

Here we have **@NgModule** decorator that we are importing from the @angular/core. It tells what belongs to our app.

**Using template and styles :**

The templateUrl is replaced by the **template**