

Angular Application Setup

First Time Installation

Before creating your FIRST project, you will need to make sure you have **node.js** installed. If you do not, you can download it here: <https://nodejs.org/en/download/>

If you already have node.js installed, make sure that it is up to date. If it is not, you will likely run into problems moving forward.

Update Node Package Manager by running the following in a command prompt window (*shift+right-click on Windows*)

```
> npm update
```

Also, if you have not installed an IDE, you will want to do that now. Here are two free IDE's you can download and install that work on both Mac and Windows.

- [Visual Studio Code](#)
- [Atom](#)

New Project Setup

1. Create a **root directory** for your site somewhere on your file system.
2. Drop the attached files in your site's **root directory**.
 - **config (directory)** – configurations files for your project
 - **helper.js** - shared logic for configurations
 - **webpack.common.js** - common logic for Webpack configuration
 - **webpack.dev.js** - Webpack's configuration for development environment
 - **webpack.prod.js** - Webpack's configuration for production environment
 - **src (directory)** - source code files of your project will live
 - **index.html** - serves as the containing page for your single page application
 - **main.ts** - the entry point for your application, tells application what to do after loading
 - **polyfills.ts** - various utilities to be loaded based on the running environment
 - **vendor.ts** - 3rd party references to be loaded such as Angular

Note that index.html, main.ts, polyfills.ts and vendor.ts are all referenced in webpack.common.js

- **git.ignore** - tells Git source control not to include particular files and folders
 - **package.json** - tells Node Package Manager what dependencies are needed for the project
 - **README.md** - offers instructions for humans on how to setup and running the application
 - **tsconfig.json** - configurations for the TypeScript compiler
 - **webpack.config** –pointer to configuration files needed by Webpack
3. Open you IDE and browse to your **root directory**.
(File > Open Folder)
 4. Open a command line from your IDE.
(View > Integrated Terminal)
 5. Run “**npm install**” to download your project's dependencies from Node Package Manager (as defined in package.json)

```
> npm install
```

6. Create an **app directory** under your **root directory**. This will be the home for all your JavaScript, CSS and HTML partial files.
7. Inside the **app directory**, create the following files.
 - **app.component.ts** - root component for your application
 - **app.component.html** - HTML template for your root component
 - **app.component.css** - styles for your root component
 - **app.module.ts** - root module for your application
8. Open **app.component.ts** and add the following code to define your root component.

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

@Component({
  selector: 'app',
  templateUrl: 'app.component.html',
  styleUrls: [ 'app.component.css' ]
})

export class AppComponent {
}
```

9. Open **app.module.ts** and add the following code to define your root module.

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

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

@NgModule({
  imports: [ BrowserModule ],
  declarations: [ AppComponent ],
  bootstrap: [ AppComponent ]
})

export class AppModule {
}
```

10. Open **main.ts** and add the following code to initialize Angular and bootstrap your root module.

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { AppModule } from './app.module';

const platform = platformBrowserDynamic();
platform.bootstrapModule(AppModule);
```

11. Open **index.html**. In the body tag, add an **app** element that matches your selector in **app.component.ts**.

```
<app>Loading...</app>
```

12. Open up **app.component.html**. Add some “Hello World” content to be shown when the application runs.
13. You are now ready to start your application. In your IDE terminal window, type “**npm run dev**”.

```
> npm run dev
```

14. Open your browser and go to <http://localhost:3000> (the starting URL defined in configuration). If all was setup correctly, you should see the “Hello World” content you added to your **app.component.html**.

Congratulations on creating your first Angular app!