**Bootstrapping in Angular**

**What is a Bootstrapping**

Bootstrapping is the process of initializing or loading our Angular application.

Angular takes the following steps to load our first view.

1. Loads Index.html
2. Loads Angular & Third-party libraries & Application
3. Executes application entry point (main.ts)  contains the bootstrap code
4. Load & execute Root Module (app.module.ts) The framework then processes the metadata defined in AppModule and initializes the application accordingly. It contains declaration of components , directives, services … used in application
5. Executes the Root Component (app.component.ts)
6. Displayes the template (app.component.html)

**Index.html loads first**

Web apps need a starting point. Index.html is usually the first page to load. Let us open the file and find out what it contains. You will find it under the src folder.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <!doctype html>  <html lang="en">  <head>    <meta charset="utf-8">    <title>GettingStarted</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> |

There are no javascript files in the index.html. Neither there is a stylesheet file. The body of the files has the following HTML tag.

|  |  |
| --- | --- |
| 1  2  3 | <app-root></app-root> |

How do Angular loads?  To find out, let us build our application

**Building Application**

To run our application, we use the [Angular CLI](https://www.tektutorialshub.com/angular/angular-cli-tutorial/) command ng serve.

ng serve builds our application but does not save the compiled application to the disk. It saves it in memory and starts the development server. ng serve also watches over the project. If we make any changes to the application, it will re-compile and update the file.

To view the compiled application, we need to build the project. We can do that using the ng build. Open the command prompt and run the ng build command. This will build and copy the output files to the dist\getting-started folder (Older versions copied the files to the dist folder).

|  |  |
| --- | --- |
| 1  2  3 | ng build |

Now open the dist\getting-started and open the index.html.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <!DOCTYPE html>  <html lang="en">    <head>      <meta charset="utf-8" />      <title>GettingStarted</title>      <base href="/" />      <meta name="viewport" content="width=device-width, initial-scale=1" />      <link rel="icon" type="image/x-icon" href="favicon.ico" />      <link rel="stylesheet" href="styles.56e3e15f5d5be678.css" />    </head>    <body>      <app-root></app-root>      <script src="runtime.87dd532f2ee308a5.js" type="module"></script>      <script src="polyfills.96a2a660c10c5395.js" type="module"></script>      <script src="main.6ea3cc439cb6e266.js" type="module"></script>    </body>  </html> |

You can see that the compiler included four script files. They are runtime, polyfills, styles, & main.

runtime.js: Webpack runtime file  
polyfills.js – Polyfill scripts for supporting the variety of the latest modern browsers  
styles.js – This file contains the global style rules bundled as a javascript file.  
vendor.js – contains the scripts from the Angular core library and any other 3rd party library.  
main.js – code of the application.

The Webpack module loader adds these files.

**What is Webpack?**

[Webpack](https://webpack.js.org/) is a bundler. it scans our application looking for [JavaScript](https://www.tektutorialshub.com/javascript-tutorial/) files and merges them into one ( or more) big file. [Webpack](https://webpack.js.org/) has the ability to bundle any kind of file like JavaScript, CSS, SASS, LESS, images, HTML, & fonts, etc.

The [Angular CLI](https://www.tektutorialshub.com/angular/angular-cli-tutorial/) uses Webpack as a module bundler. Webpack needs a lot of configuration options to work correctly. The Angular CLI sets up all these configuration options behind the scene.

The Webpack traverses through our application, looking for [JavaScript](https://www.tektutorialshub.com/javascript-tutorial/) and other files, and merges all of them into one or more bundles. In our example application, it has created five files.